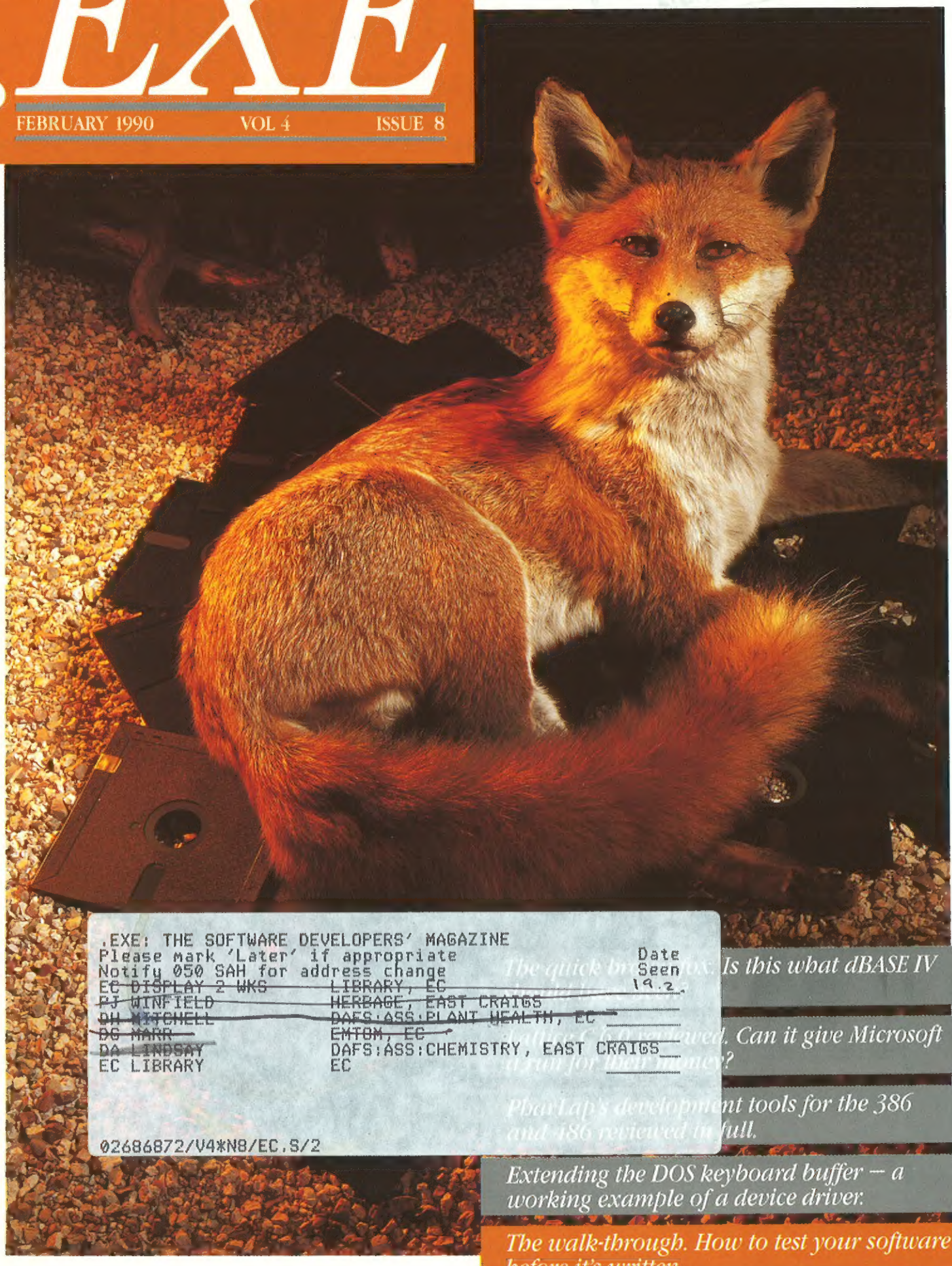
.EXE

FEBRUARY 1990 VOL 4 ISSUE 8

4TH BIRTHDAY ISSUE

The Software Developers' Magazine



The quick brown fox. Is this what dBASE IV

dBase C reviewed. Can it give Microsoft a run for their money?

PharLap's development tools for the 386 and 486 reviewed in full.

Extending the DOS keyboard buffer — a working example of a device driver.

The walk-through. How to test your software before it's written.

THE TRAINING LEADERS

To 'C' PROGRAMMERS AT ALL LEVELS

"The best path for complete 'C' training begins at QA".

'C' has become the dominant language in computer programming and provides access to the software industries major application program interfaces.

'C' is probably now, or shortly will be, your way forward in program development – and QA can really help you along that path.

Only QA's extensive 'C' curriculum totally supports the training needs of all 'C' programmers, from beginner through to highly experienced professional. Each course can stand alone or can be entry point to reach any level of skill or knowledge you may set yourself.

In common with all QA courses, our instructors are experienced professionals themselves, our training materials are perfected for efficient learning, and all our own skill and energy is channelled toward helping you get the maximum benefit from your training.

Send for more details today or call Samantha Trinder on 0285 655888.

BEGINNERS START HERE

OTHERS START HERE...

...OR ANYWHERE HERE...

...INCLUDING C++

'C' COURSE OPTIONS

C PRIMER
2 Days £340

HLL concepts via C ● Overview of C programming environment ● Developing simple C programs ● Data typing, block structure, scope concepts ● Introducing C functions and modular programming

C PROGRAMMING
4 Days £680

C programming structure and style ● Data types and variables ● Flow of control in C programs ● Pointers and address manipulation ● Portability and efficiency considerations ● Debugging C programs ● Working with larger C programs

C TECHNIQUES FOR OS/2 & P.M.
1 Day £160

Using C in an OS/2 environment ● PC and OS/2 extensions to C ● Mixed Models of compilation ● Advanced C techniques for OS/2 ● The OS/2 API C interface ● Dynamic link library concepts ● OS/2 development tools overview

C TECHNIQUES FOR DOS
1 Day £160

Implications of using C under DOS ● PC and DOS extensions to C ● Models of compilation and segmentation ● Interfacing to DOS and BIOS from C ● Linking C to other languages ● DOS development tools overview

C ADVANCED SOFTWARE DEVELOPMENT
4 Days £720

A lightning tour of C ● Advanced pointer techniques ● 'Defensive' programming techniques in C ● Advanced C concepts ● A closer look at C's library routines ● Recent ANSI and SAA C developments ● C development tools overview

C STANDARDS
1 Day £200

C standards – aims, goals and benefits ● Defining practical and realistic standards ● Naming and layout standards ● C program organisation standards ● Defensive C programming ● Portability, maintainability and efficiency

C++ PROGRAMMING
4 Days £720

Object orientated programming concepts ● OOP and C++ ● Concept of classes ● Class hierarchys, derivation and inheritance ● Constructors and Destructors ● Application examples

Please complete and mail to QA Training, Cecily Hill Castle, Cirencester, Glos GL7 2EF.

☐ Tick here for full details on all our 'C' courses.

Send me details on (tick as appropriate):

☐ C Primer ☐ C Programming ☐ C Techniques for OS/2 & PM
☐ C Techniques for DOS ☐ C Advanced Software Development
☐ C++ ☐ OS/2 Courses ☐ Windows Courses ☐ Support Courses

NAME | TITLE

COMPANY

ADDRESS

TEL. NO.

QA TRAINING

CIRCLE NO. 740

Vol 4 Issue 8 .EXE February 1990

Editor
Robert Schifreen

News Editor
Will Watts

Editorial Assistant
Wendie Leek

Advertising Manager
Philip Northcott

Sales Executives
Sandra Inniss-Palmer
Jeremy Cox

Publisher
Jon Howell

Reader Services & Promotions
Helena Adams

Design & Layout
Katerina Adams
Rachel Affleck

Office Administrator
Carla Pearce

Cover Photography
Richard Cook

Editorial Policy
We aim to provide news, product reviews and technical features for those who develop PC software for both commercial sale and internal company use. Our policy is not to review any software product until it is available in its final form, in order to provide accurate figures on code size and speed. The Magazine welcomes articles from readers – please ask for our contributor's guide.

Subscriptions
.EXE is a magazine for managers and programmers involved in the development of software for the IBM PC and compatibles. .EXE is available only on subscription, which costs £35 per year (11 issues) in the UK. The Magazine is published on the 26th of the month preceding the cover date. There is no January issue. International subscription rates can be found on the subscription card in this issue. Back issues are available at £3.50 each. To subscribe, call 044282 4501. We can invoice your company or take VISA or ACCESS cards.

.EXE Magazine is independent and not affiliated to any vendor of hardware, software, or services. Published by Process Communications Ltd. 10, Barley Mow Passage, London W4 4PH, England. Telephone 01 994 6477

ISSN 0268-6872

Reacting to Readers
We aim to provide high quality facts and information, about subjects relevant to readers. Our annual Reader Survey gives us an insight into the tools, methods language and hardware that is being used, and that will be used, in the process of software development.

We aim to keep in touch with software developers as much as possible. For this reason, we hold twice-yearly meetings at the .EXE offices and around the country, where readers can talk directly to the editorial staff and suggest ways in which the Magazine could be improved.

The Hunt

Shortly before Christmas last year, I decided that I'd treat myself to a record. Six months before, I'd seen a programme on ITV called Folk Aid for Lockerbie, in which a number of Scottish folk singers performed at a fund-raising event. Being a fan of Scottish and Irish music, one of the songs had taken my fancy, and I was determined to get a recording of the singer singing it. Unfortunately, I could not remember the singer's name, and six months is a long time in television.

So, to set about getting the record. I started by scouring catalogues in a couple of local record shops, but to no avail. I called London Weekend Television and Thames TV, one of which had screened the program (I didn't remember which). They suggested that I call Grampian TV, who in turn suggested that it must have been Scottish TV that made the program. It was neither of these.

Next, to TV Times Magazine, who were very helpful. A nice lady told me that yes, they did have back issues of TV Times, and yes, the listings in TV Times would almost certainly include the name of the production company. Unfortunately, the library wasn't open to the general public but, if I knew the date on which the programme went out, she would look it up for me, and by the way had I tried Border TV in Carlisle? No, I hadn't, but I did.

A couple more phone calls later, I discovered that it was indeed Border TV that had actually made the programme. I was even given the name of the director. I tried calling him a couple of times, but he was busy. In the end, I typed a Fax message, printed it out and went next door to send it. The next afternoon, a reply came. I now know the singer's name. I can now look up his discography in another catalogue, at a local record shop. They will then be able to order the record for me, I hope.

Another recent quest for information concerns my search for technical data regarding smooth pixel scrolling on EGA and VGA monitors. I was struggling with all the textbooks, and wanted more details. There were two places I knew that would have information. Sitting at my office PC, I typed a single command. This scanned through 2 years' of .EXE Magazines, which I have on my hard disk. It pulled out a couple of references, but I really wanted more. I changed from drive C to drive E. This is my CD-ROM drive, which contains Computer Library. This essential accessory contains a year's worth of computer magazines from the US and the UK. In the 550 MB of data on the disk is the full text of around 30 titles, and extracts from around 120 more. I asked the system to search for all ocurrences of SMOOTH and SCROLL within the same sentence and, within half a minute, I had the titles of dozens of suitable articles. A couple of menu selections later, I had the full text of 2 detailed articles, along with accompanying program listings, copied to my own hard disk in plain ASCII format. I could now search the files, and read the information that I was looking for.

So far, my search for the Scottish folk record has taken a couple of weeks of intermittent phone calls, visits to record shops and harassing busy people. My record player is still hungry. My pixel scrolling routines, on the other hand, are coming along well. The search through a total of almost 1800 magazines took about 20 minutes, and there are lots more dial-up databases I could have tried if my local selection had not had the information I required.

Ten years ago, various big companies tried to set up computer databases for public access. There would, we were told, be a computer terminal in every home by 1990 or so. There would be no need for printed telephone directories, trips to the grocer and so on. The French have managed it, to a certain extent, but Britain has not. If the Information Revolution is really going to happen, and be accessible by more than just a lucky few, it will be up to software developers like us to bring it about. There is, though, still much to be done. Meanwhile, if you have a listing of Archie Fisher's recordings, please drop me a line.

THE C LANGUAGE

Hich C V1.6 has been considerably improved, with better Microsoft C compatibility, and new documentation

C COMPILERS

Avocet C	PC-DOS	£210
Aztec C86 Professional	MS-DOS	£120
Aztec C86 Developer	MS-DOS	£175
Aztec C86 Commercial	MS-DOS	£290
CI C86 Plus v1.3	MS-DOS	£290
ECO-C88 V4.0	MS-DOS	£ 70
HIGH C (Metaware) v1.5	MS-DOS	£420
HIGH C 386 v1.5 PL386&MS-DOS		£635
Hi-Tech C	MS-DOS	£135
Lattice C v6 OS/2&MS-DOS		£175
Mark Williams LETS C v4	MS-DOS	£ 60
Microsoft C v5.1 OS/2&PC-DOS		£250
Microsoft QuickC v2	PC-DOS	£ 60
M'soft QuickC/Quickasm	PC-DOS	£105
MIX C	MS-DOS	£ 20
MIX Power C & lib s'ce	PC-DOS	£ 29
Turbo C v2	PC-DOS	£ 79
Turbo C Professional v2	MS-DOS	£160
Topspeed C for OS/2	OS/2	£335
Topspeed C (std)	PC-DOS	£135
Topspeed C Professional	PC-DOS	£270
Watcom Express C v7	PC-DOS	£ 65
Watcom v7 opt comp + tls	PC-DOS	£250
Watcom C/386 PL386&MS-DOS		£625
Zortech C v3	PC-DOS	£ 45
Zortech C++ v2	PC-DOS	£120
Z'tech C++ v2 Developer	PC-DOS	£270
Aztec C86 Developer	CP/M-86	£245
Aztec C86 Personal	CP/M-86	£140
Hi-Tech C	CP/M-86	£135
Lattice C	CDOS	£395
HIGH C (Metaware) Flex OS 286		£420
Avocet C	CP/M-80	£210
Aztec C Personal 1.06D	CP/M-80	£120
Aztec Commercial 1.06D	CP/M-80	£200
Eco-C w. Lib s'ce Z80 +	CP/M-80	£ 60
Hi-Tech C Z80 +	CP/M-80	£ 99
Mix C Z80 +	CP/M-80	£ 35
Toolworks C/80 v3.1	CP/M-80	£ 45
Aztec C65 V1.05 Apple DOS		£200
Aztec C65 V3.20 Apple PRO-DOS		£245
Aztec C68K MPW C	MACINTOSH	£110
Aztec C68K C	MACINTOSH	£ 75
Aztec C68K/AM Developer	AMIGA	£175
Aztec C68K/AM Prof'snl	AMIGA	£120
Aztec C68K/ST Developer	ATARI	£175
Aztec C68K/ST Prof'snl	ATARI	£120
Hi-Tech C	ATARI	£ 99
Lattice C	ATARI	£ 85
Mark Williams C v2	ATARI	£110
Laser C	ATARI	£135
Prospero C	ATARI	£105

C CROSS COMPILERS

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020. Please call for information or advice.

C INTERPRETERS

Instant C/16M supports large programs on 286/386 machines.

C-terp V3.0X	PC-DOS	£165
Interactive C	PC-DOS	£195
Introducing C	PC-DOS	£ 85
Living C Plus	PC-DOS	£135
Instant-C v3	PC-DOS	£380
Instant-C/16M	PC-DOS	£680
RUN/C Professional	MS-DOS	£110

PROLOG LANGUAGE

Arity Standard	PC-DOS	£ 79
Arity Prolog Int. v5.1	PC-DOS	£225
Arity Int. & Comp. v5.1	PC-DOS	£495
Flex Expert Dev Sys	PC-DOS	£1210
LPA PROLOG Prof v3	PC-DOS	£755
Micro-PROLOG v3.1	MS-DOS	£ 70
Prolog-86 plus v1.0	MS-DOS	£195
Prolog-2 Personal	PC-DOS	£145
Prolog-2 Programmer	PC-DOS	£495
Prolog-2 Professional	PC-DOS	£995
Turbo-Prolog v2	PC-DOS	£ 79
ADA Educ.Prolog	MS-DOS	£ 26
ADA FS Prolog	MS-DOS	£ 45
ADA VMI Prolog	MS-DOS	£ 70
ADA VML Prolog	MS-DOS	£150
Micro-Prolog v3.1	CP/M-86	£ 70
Micro-Prolog v3.1	CP/M-80	£ 70
Prolog-1 v2.2	CP/M-80	£225
LPA Mac Prolog v3 Apple Mac Pls		£450

C LIBRARIES

Microsoft OS/2 Presentation Manager Toolkit is now in stock.

DATABASE

Btrieve	OS/2	£400
Btrieve v5	MS-DOS	£170
Btrieve/Network	MS-DOS	£400
XQL	OS/2,PC-DOS	£540
Xtrieve Plus	OS/2,PC-DOS	£400
C/Database T'lchest sce	PC-DOS	£ 39
CBTREE	(source any C)	£135
C-Index/Plus	(source any C)	£175
C-ISAM (L,MS)	MS-DOS	£220
C-tree	(source any C)	£230
Essential Btree (s'ce)	PC-DOS	£120
R-tree	MS-DOS	£170
db-VISTA III s'gle user	PC-DOS	£575
db-Query v2 s'gle user	PC-DOS	£575
Lattice dBCIII (L,MS)	MS-DOS	£165
Lattice dBCIII+ (L,MS)	MS-DOS	£385
SftFcus Btree&Isam (sce any C)		£ 80
Virtual Memory File Man.	PC-DOS	£ 80

GRAPHICS

Enhanced Graphics Tlkt	PC-DOS	£170
Essential Graphics v2	PC-DOS	£175
GFX Graphics	PC-DOS	£ 75
GraphiC v5 (CI,L,DS,MS)	PC-DOS	£290
Graphics Menu (MSC,TC)	PC-DOS	£105
GSS Kernel system	PC-DOS	£350
GSS Graph Dev Toolkit	PC-DOS	£350
GSS Graphic Plotting sys	PC-DOS	£245
HALO '88 (MSC5,LAT,TC)	PC-DOS	£230
HGraph (MSC)	PC-DOS	£ 80
Ingraf (MSC)	PC-DOS	£175
MetaWINDOW (CI,L,MS,T)	PC-DOS	£145
MetaWINDOW/Plus	PC-DOS	£190
MetaWINDOW/Premium	PC-DOS	£315
TurboWINDOW/C (Turbo)	PC-DOS	£ 75
QuickWINDOW/C (QC)	PC-DOS	£ 75

SCREEN & WINDOWS

Vitamin C (MS5.1) OS/2&	PC-DOS	£240
Power Screen (MS&TC)	PC-DOS	£105
Blaise View Mngr.(s'ce)	PC-DOS	£245
Lattice Curses w s'ce	PC-DOS	£130
Entelekon Windows(s'ce)	PC-DOS	£105
Multi-windows (MS,L)	PC-DOS	£220
Panel Plus II (source)	PC-DOS	£270
Vitamin C (source)	PC-DOS	£165
Vermont Views (MSC,TC)	PC-DOS	£275
Greenleaf Datawindow	PC-DOS	£180
Greenleaf Makeform	PC-DOS	£ 60
C-Scape v3.1 (MS,L,TC)	PC-DOS	£330
Zortech Windows(ZC++ZC3)	PC-DOS	£ 39
Curses/PC w. s'ce	PC-DOS	£265
D-Tree (source)	PC-DOS	£345
Greenleaf Functions s'ce	PC-DOS	£105

GENERAL FUNCTIONS

C+0 Class Lib (MS,TC)	PC-DOS	£155
C+0 Cl's Lb (MS)WINDOWS&	PC-DOS	£155
C/Utilities T'chest sce	PC-DOS	£ 39
CQL s'ce	PC-DOS	£270
C-Worthy Int.Lib (L,MS)	MS-DOS	£175
Code Runner	PC-DOS	£105
G'leaf Super Functions	PC-DOS	£135
G'leaf Business Mathlib	PC-DOS	£180
Smorgasbord C-food	PC-DOS	£ 65
C Tools Plus/6.0 (MS)	PC-DOS	£105
Power Search	PC-DOS	£105
Turbo C Tools v2	PC-DOS	£105
Essential C Utils s'ce	PC-DOS	£110
Entelekon Funct(source)	PC-DOS	£105
Pro-C & Workbench	PC-DOS	£350
Resident-C (MS,L)(s'ce)	PC-DOS	£120
The Heap Expander(MS&TC)	PC-DOS	£ 65
Zortech Supertext	PC-DOS	£ 39

COMMS LIBRARIES

C Asynch Manager V3s'ce	PC-DOS	£135
Essential Comms(L,MS,T)	PC-DOS	£145
Greenleaf Comms(source)	PC-DOS	£135
Multi Comms (L,MS)	PC-DOS	£165
Silvercomm C Asynch	PC-DOS	£145
Zortech Comms (QC&TC)	PC-DOS	£ 65
Zortech Comms(ZC++&ZC3)	PC-DOS	£ 65

SCIENTIFIC LIBRARIES

Wiley Scientif.Lib.v2	ANY C	£155
Mathpak 87 (L,MS)	MS-DOS	£ 85

PROGRAMMERS UTILITIES

PC-Lint OS/2,	MS-DOS	£ 85
C-Documentor	PC-DOS	£195
C-Scan	PC-DOS	£195
C Toolset	MS-DOS	£ 90
Clear+ for C	PC-DOS	£165
Lattice Comp Companion	MS-DOS	£ 65

REAL TIME & MULTI-TASKING TOOLS

Concurrent C (PC/MPX)sce	MS-DOS	£ 60
Multi C (L,MS,CI,TC)	PC-DOS	£165
Op.Sys.T'box (MS) s'ce	PC-DOS	£ 65
Timeslicer v3.01 (L)	PC-DOS	£175
Timeslicer v5 (MS)	PC-DOS	£175
Over-C (L,MS)	PC-DOS	£225

PROGRAMMING TOOLS

Ada Compilers	Algol Compilers
Assemblers & Libs	AWK
Basic Compilers	Basic Interpreters
Basic Utilities	Basic Libraries
BCPL Compilers	C Compilers
C Interpreters	C Libraries
C Utilities	Cobol Compilers
Comms.Libraries	Cross Assemblers
Database Libs.	Debuggers
Dis-assemblers	Editors
Engineers Libs.	Expert Systems
Forth	Fortran Compilers
Fortran Libraries	Graphics Libraries
Icon	Linkers/Locaters
Lisp	Modula-2
Nial Interpreters	OPS 5
Pascal Compilers	Pascal Libraries
Prolog	Rexx
Risc	Screen Libraries
Smalltalk	Snobol

We stock many items for which there is no space in these advertisements.

PRICES & DELIVERY

Prices do not include VAT or other local taxes but do include delivery in UK and Europe. Please check prices at time of order, ads are prepared some weeks before publication.

This page lists some of our products. Call us for a complete pricelist.

Order by phone with your credit card.

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, ISIS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, DEC RT-11, IBM BEF, ATARI ST, AMSTRAD, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt. For more information call us.

ADA COMPILERS

IntegrAda - The first Ada compiler with an easy to use Turbo-like environment.

Janus/Ada compiler	PC-DOS	£ 265
IntegrAda	PC-DOS	£ 645

ASSEMBLERS

The Turbo Assembler/Debugger from Borland is in stock. The new PharLap 386/VMM uses the virtual memory features of the 386 to allow very large applications.

MS Macro-86 v5.1 OS/2&	PC-DOS	£ 95
PharLap 386 ASM.LINK	MS-DOS	£400
PharLap 386 ASM/Linkloc	MS-DOS	£565
PharLap 386/VMM	MS-DOS	£240
Optasm	MS-DOS	£ 95
Turbo Assembler/Debugger	PC-DOS	£105
2500AD Z80 ASM	CP/M-80	£ 70
Dig.Res. RMAC	CP/M-80	£180
SLR Z80ASM	CP/M-80	£ 40
SLR Z80ASM-PLUS	CP/M-80	£140
SLR MAC	CP/M-80	£ 40
SLR MAC-PLUS	CP/M-80	£140
SLR 180 (Hitachi)	CP/M-80	£ 40
SLR 180-PLUS (Hitachi)	CP/M-80	£140

Not all assemblers are supplied with a linker. Check before ordering.

News

Latest Microsoft BASIC

Microsoft has launched BASIC 7.0, which is now known as the Professional Development System rather than QuickBASIC. As before, there's a QuickBASIC (now called QuickBASIC Extended) environment in which to write and debug programs, and a command-line compiler that produces final .EXE files. The product costs £335, and will not replace the £70 QuickBASIC 4.5.

Research by Microsoft in the US found that around 62% of QuickBASIC users are producing software that will be distributed commercially. This, apparently, is the reason for the host of new additions to the product and, I suppose, why the SETUP program says that you need 14 MB of free hard disk space before installing.

Major additions include: a text-based windowing and menu system, supplied in source code form that you can customise if you wish; a presentation graphics toolbox; an ISAM toolbox for writing database applications; an optimising compiler; EMS support for your programs, and for the run-time environment; support for creating 16 MB executable files via the use of overlays; string data moved to far memory to avoid DGROUP overcrowding; over 2000 pages of printed documentation in addition to online help; the ability to produce OS/2 applications.

Incidentally, the original Microsoft press information contained an error. If you see any stories suggesting an RRP of £495, this is incorrect.

Sourcer updated

Sourcer, the wonderful disassembler that we mention occasionally on these pages, is now up to version 2.11. The latest edition handles the full 386 and 486 instruction set in both real and protected modes, and now makes up to 9 passes on a file, rather than 5. The latest version, known as Sourcer 486, also includes the BIOS pre-processor as standard. This utility looks through your BIOS ROM, working out the entry points for all the standard routines. This data is then added to a file of pre- written comments which, combined with a run through the Sourcer program itself, provides you with a complete, commented disassembly of your BIOS. Call V Communications for more details, on 0101 408 296 4224. The Company has also just launched UNPACKER, which is claimed to undo the effects of the EXEPACK utility.

Assembly Library

POWER LIB is a new function library for assembly language programmers. There are over 250 routines, in categories including drop-down menus, screen functions, dates, times, file encryption, string handling, sound effects and user input. The company reckons that a fully functional program with nested windows, shadows and menus can fit in less than 5K of code. The product comes from Quantasm, who produce the AsmFlow tool for turning assembly language source code into flow charts. POWER LIB costs $100, or $300 with complete assembly language source code. It works with MASM, Turbo Assembler or OPTASM, though links to other languages are scheduled for release within the next couple of months. Credit card holders can call Quantasm on 0101 408 244 6826.

ANSI nearer

ANSI has moved a step nearer to publishing its notoriously delayed C standard. Russell Hansbury, an unhappy programmer, who, by making a series of complaints, had succeeded in entangling the relevant ANSI committees in yards of their own red tape, has dropped his appeal against the draft standard. The way to publication now seems clear, although one cynical standards-watcher has pointed out that there still remains one more avenue open to Hansbury to bring matters grinding to a halt .

This is still not the end of C's tortuous standardisation process, however. At the last international meeting to discuss language standards, although the ANSI draft was supported, it was decided to start work on an *updated* version of the standard. This was because it was felt that the current document contained too many ambiguities and omissions: the UK delegation alone had a list of 35 complaints. The new standard is to be produced at an international level (rather than American), presumably with the intention of moving along rather more nimbly.

No need to worry about the portability of your code under the new (version two) standard, however. The ISO committee will not be devising new spellings for `printf()`. An example of a question to be addressed in the forthcoming document: 'What do you get if you compile and run a program without a `main()` function?' .EXE's suggested answer: what you deserve.

dBASE Add-Ons

A rather useful .DBF data file has appeared on Nantucket's dial-up technical support system recently. Called DBTUT, it's a rather large (over 100K) database of all known third-party add-ons for Clipper. The products are primarily US based, but it's still a useful reference. Nantucket should be able to supply the file; contact them on 0707 373600 (note the new number – they have recently moved).

Got it taped

If you find reading through all those trade newspapers too time- consuming (.EXE excepted, naturally), then how about a new eyes-free solution from Dutch publisher Publis. CompuTape is biweekly computer news on a cassette tape, read out in the manner of a radio news bulletin. The specimen sent to us was actually quite boring, but you can't have everything. More details on 010 31 3465 63997.

Flight Simulator 4.0

Version 4 of the Microsoft Flight Simulator is now available. Most exciting new feature is an aircraft designer facility, which lets you change almost every dimension of a plane (wing span, rudder size etc), as well as the propulsion method. There's also a dynamic weather generator, which helps you crash more easily. RRP is £45.

Paradox Engine

Borland is soon to release a library of C functions which will let you read and write to databases produced by the company's Paradox product. The Paradox Engine will contain over 70 function calls in its API, and will be priced at £325.

Translation Service

A specialist software translation service has been launched by Tek Translation. TekSoft, as the service is called, will translate English software and documentation into most European languages. All work is carried out by computer-literate people working in their native tongue. Details from Tek Translation on 01 749 3211.

Learn yourself C

'The Intensive C Course' is based on ten 30-minute videos containing material from degree courses given at Birkbeck College, London. A new edition of the videos has just been published by Chartwell-Bratt, priced £850. Call 01 467 1956 for more information.

Borland's second loss

Following the recent announcement that it had abandoned Turbo BASIC, Borland has now revealed that it has also transferred the development and marketing rights of Turbo Prolog. The new owner is a Danish software house called the Prolog Development Centre, which was the original developer of the product.

At PDC, Turbo Prolog, now renamed PDC Prolog, continues to evolve. There is already a new MS-DOS release (V3.2) and an OS/2 version should appear in February. The company is also working on SCO UNIX and object- oriented versions, intended for release later in the year. PDC is currently looking for a UK distributor, so for the time being you can only buy the software direct from Denmark (phone 010 45 3672 1022, they all speak English), priced $250 US for the MS-DOS package and $600 for OS/2. Existing registered Turbo Prolog owners will receive a letter from PDC, offering them an upgrade for $80.

Meanwhile, ex-stablemate PowerBASIC (nee Turbo BASIC) is now being marketed by a Californian company called Spectra Publishing (0101 408 730 9291). PowerBASIC V2.0 features many enhancements over Turbo BASIC V1.1: there are five new variable types, including fixed and floating point BCD decimals, there is the facility to define data structures 'on the fly' during program execution, the 64 KB limit for string space is broken and there are some powerful new commands for array manipulation. Spectra is also in the throes of arranging UK distribution so, if you are anxious to get hold of the software, you should call them in the US.

PowerBASIC is priced at $154.95 (including $25 shipping), with a $20 discount if you order before 1st March. Upgrades from Turbo BASIC V1.1 are available for $84 inclusive – call Spectra with your TB serial number to get one.

Remote control

Remote Keyboard is a two-component package: a receiver, which plugs into the serial port, and a battery powered transmitter, which closely resembles a TV remote control channel changer. The system is primarily intended for lecturers giving presentations on PCs. Remote Keyboard frees them from the inconvenience of having to start typing halfway through a declamation. I dare say it has other uses as well. The package costs £349. The UK distributor is the AZCOM Marketing Company (0932 851972).

BCS OOSE

The British Computer Society is calling a conference to try to harness UK OOP expertise. Delegates attending the conference will contribute a written account of their OOP experience several weeks before the event: recommended books, useful rules of thumb, good courses and so on. There will be further opportunities to offer your ha'penny's worth on the day, which is 9th March. The intention is to collect all the thoughts into a book. The conference will be held at the Strand Palace Hotel, London. Call Karyn McCartney on 01 637 0471 to find out more.

TopSpeed

Jensen & Partners International has finally released its long awaited TopSpeed C product. The new compiler can generate code for both MS-DOS and OS/2, boasts an integrated environment that is good enough to impress even hardened Turbo users like your correspondent, includes Borland compatible graphics libraries and, according to the manufacturers, contains a particularly fierce optimiser. The package is available in three editions: Standard (£149), DOS Professional (£295) and OS/2 Professional (£370). You can order direct from the manufacturers (01 253 4333). We will be reviewing TopSpeed C in the near future.

JPI is now beta-testing Version 2.0 of its Modula-2 compiler. The new release of TopSpeed Modula-2 will have the same environment as the C compiler (nine edit buffers, 500 KB per file, context-sensitive language help etc) plus various new language features. You can now generate function calls using C parameter passing conventions, so the Modula-2 programmer, previously somewhat isolated, will have access to the large range of existing C libraries. Inevitably, JPI has also devised some object-oriented extensions. These appear to be similar to Borland's Turbo Pascal V5.5: there is a new `CLASS` keyword (equivalent to TP's `OBJECT`), methods may be implemented as static or virtual, and objects may be allocated either as statics or in heap space using `NEW` and `DISPOSE`. The descriptive literature also promises, somewhat mysteriously, 'facilities...for manipulating the method table of an object'. Sounds interesting. TopSpeed Modula-2 V2.0 is scheduled for release in February; the pricing structure will be the same as for TopSpeed C.

QuickBASIC for UNIX

Basmark Software have produced a clone of Microsoft QuickBASIC that runs under various versions of UNIX. If you need to produce software quickly, that runs under DOS and UNIX, and you don't like C, this could be a solution. Prices vary from £850, for the SCO Xenix version, to £2450 for a Sun 4. Call the European office on 0734 791737 for details, or go direct to the US on 0101 216 621 7650.

LISP Conference

EUROPAL'90 is the first European Conference on the Practical Application of LISP. The conference is to be held on 27th-29th of March in Cambridge (UK). Call the organisers at Applied Workstations Ltd on 0306 889485 for a brochure and, if you're lucky, a nice blue pencil like the one they sent me.

SCSI for OS-9

Microware has announced a 680x0 SCSI Driver Pak for its OS-9 real time operating system. The 'Pak', which consists of high-level device drivers, a low-level SCSI module and device descriptors, includes source code listings of all software. This allows optimisation of the code for the target system. Microware may be contacted on 0489 886699.

Helios Graphics

Bristol based Peribelion Software (0454 612777) has produced a new graphics library for Helios, its transputer operating system. Programs running under Helios on PC cards can now drive CGA, EGA and VGA graphics boards. The software, which costs £300, is source compatible with the Microsoft C graphics library, to aid swift porting to transputer.

Imperfect 486

Desperate to be at the head of the rush to ship 80486-based PCs, UK manufacturer HM System has been sending out its 486 Minstrel Workstations with bugged CPUs. To be fair, the company says that the problem in the 486 chips is very obscure, and has guaranteed to replace the faulty CPUs by the summer.

You're never alone

Our congratulations to Strand Software Technologies, which won the 1989 BCS Award for Technical Achievement for the development of the STRAND 88 parallel programming language. The language was described in an article by Professor John Florentin in the October issue of .EXE Magazine.

What now for OS/2?

IBM's OS/2 LAN Server, and Microsoft's OS/2 LAN Manager, are about to become one and the same product. Speaking at Comdex, the companies said that the move was being made to avoid confusion on the part of users. At present, around 30 of the LAN Manager API calls are missing from IBM's implementation, but this problem will soon be cleared up.

We also understand that, by the time OS/2 2.0 (the 386 version) comes around, the distinction between IBM's Extended Edition and Microsoft's normal version will be gone. There are two possible explanations for this major change in marketing policy: either IBM is trying to push more and more OS/2 responsibility onto Microsoft, or Microsoft is hassling IBM and insisting that Microsoft be given more responsibility. Rumours that have crossed .EXE's desk in recent weeks suggest that the former is nearer the truth, and that IBM is unwilling to keep spending money on developing OS/2 in order to make it appeal to potential users.

Comparing Files

Research by .EXE shows that few companies use revision control software. If you're one of those that don't, and you need to find out what's changed between two versions of a source code file, DocuComp may be the answer. Unlike the file comparison utility that comes free with most versions of DOS, DocuComp has some sophisticated algorithms for coping with complex files, including ones of different lengths.

Details from Advanced Software in Sunnyvale, California, on 0101 408 733 0745. There's no UK distributor as yet, but they're working on it.

BATPUG

BATPUG stands for the British All Turbo Pascal Users Group. It has a quarterly journal, full of tips and tricks; plus a library of public domain Turbo Pascal relevant software, which is sold to members at £2 a disk. The group is keen to recruit. If you are interested, please write to David Bolton, BATPUG, 12 Clegg Avenue, Thornton, Cleveleys, Blackpool FY5 1BJ. The annual membership is £7.50, which includes subscription to the journal and an archived disk of software.

Protool

Protool is an MS-DOS based user interface prototyping tool, produced by the Wiltshire company JRM Software Ltd. It is based around an editor/interpreter/debugging environment, which you use to develop scripts in a simple proprietary language. Text mode windows and data entry fields are created and positioned using screen painting techniques. With Protool, it is possible to knock up a dummy version of an application in a few minutes.

The idea, of course, is to simplify the process of software creation and pre-empt 'that's not what I asked for' type disputes. The customer is shown a prototype of his application. When he has agreed upon its design, he can keep a copy as a reference. The actual software is then written, with a user interface modelled on the prototype. Protool costs £175 ex VAT. Phone JRM on 0225 760251.

Pop-up books

If you're a fan of the hypertext style help provided with Microsoft's QuickC V2.0, then you may be interested in a new range of products from the US company Window Book Inc. The basic package consists of a TSR program, similar to the Norton Guides, which can be popped up at the touch of a hot-key. The software then gives you access to a cross- referenced database of text. Window Book has produced a guide for MS- DOS operation called WinDOS (priced at £29), and, perhaps of more interest to .EXE readers, a hypertext version of the Ada Language Reference Manual, called Ada-LRM (£99).

There is also a customisation service, where the company will put your manuals into its hypertext format. This is charged at £22 per 2000 character page; you can include graphic images as well. For more information contact the UK distributor Milspec Systems on 0203 670770.

Business GKS

BGUL is a library of business graphics routines, capable of producing graphs, bar charts, pie charts and the like. It is available with either a C or a FORTRAN API. What makes it different is that it creates its graphical images using standard GKS calls, so the application remains portable. BGUL is initially being offered on MS-DOS, VAX/VMS, and UNIX operating systems. Call the distributor, Scientific Software Ltd, for prices (01 864 8818).

.EXE

Hi Tech Graffiti

The most exciting new printer to be launched at Comdex was the Michaelangelo V4 Paint Jet System. It's a colour ink jet printer, but with no paper mechanism. Instead, you just wheel it against a wall, and start it going. It will produce prints of up to 5 feet by 5 feet (though it can take up to four hours), and is said to work wonderfully on fabrics and venetian blinds. The price? Around $100,000.

100 Targets

American Automation has upgraded its range of C cross-compilers to ANSI standard. Over 100 different microprocessors are supported, and the compilers include a special size optimiser for squeezing large programs into limited ROM space. Prices for the package start at £595. The company is offering various inducements to get you to try its software, including free compilation of a code sample and a free demonstration disk. Call 0993 778991.

Pascal Fractals

A new book from M&T publishers deals with one of the most popular hobby programming areas. 'Fractal Programming in Turbo Pascal' by Roger T Stevens (ISBN 1-55851-106-7, US price $29.95) covers Mandelbrot and Julia sets, plus Hilbert Curves, the Lorenz and Other Strange Attractors and the Snowflake and other von Koch Curves. You'll need Turbo Pascal V4.0 or better; the book can be obtained from retail outlets.

Security Seminar

If you're quick, you can still get along to Corporate Computer Security '90, a conference and exhibition in London from 13th to 15th of February. Around 25 exhibitors are expected to show their wares, and the conferences include sessions on risk management, access control, PC security, network security, software validation and data integrity. Details from PLF Communications on 0733 558571.

Sun graphics

Oxford based Quintec Systems has produced an object-oriented graphical toolkit for its Prolog implementation, running on Sun-3 workstations. Quintec-Vieupoint lets Prolog users set up screens containing windows, buttons, dialogues and all the other WIMP paraphernalia. Quintec is on 0865 791565.

New FoxPro

Shifting the Balance Of Power in Database Management

There's a new leader in the relational database management world. Its name is FoxPro.

FoxPro is the first and *only* microcomputer database management system that combines astonishing performance with a sleek interface of amazing power and beauty.

- FoxPro offers all the elegance and accessibility of a graphic-style interface, yet operates at the stunning speeds possible only with character interfaces.
- FoxPro is so easy to learn and use, even beginners can become productive immediately; yet it's powerful and sophisticated enough to satisfy the needs of the most demanding developers and power-users.
- FoxPro gives you choices instead of limits: use a mouse or a keyboard; type commands or use the object-oriented interface; run in one window, or hundreds.
- FoxPro is so efficient, it runs in a 512K PC-XT, yet it's able to take advantage of the speed, expanded memory and extended video modes of the most advanced machines available. You don't even need a graphics card or special windowing software.

Nothing is Faster

Fox Software products are famous for their unmatched execution speed. FoxPro extends that tradition.

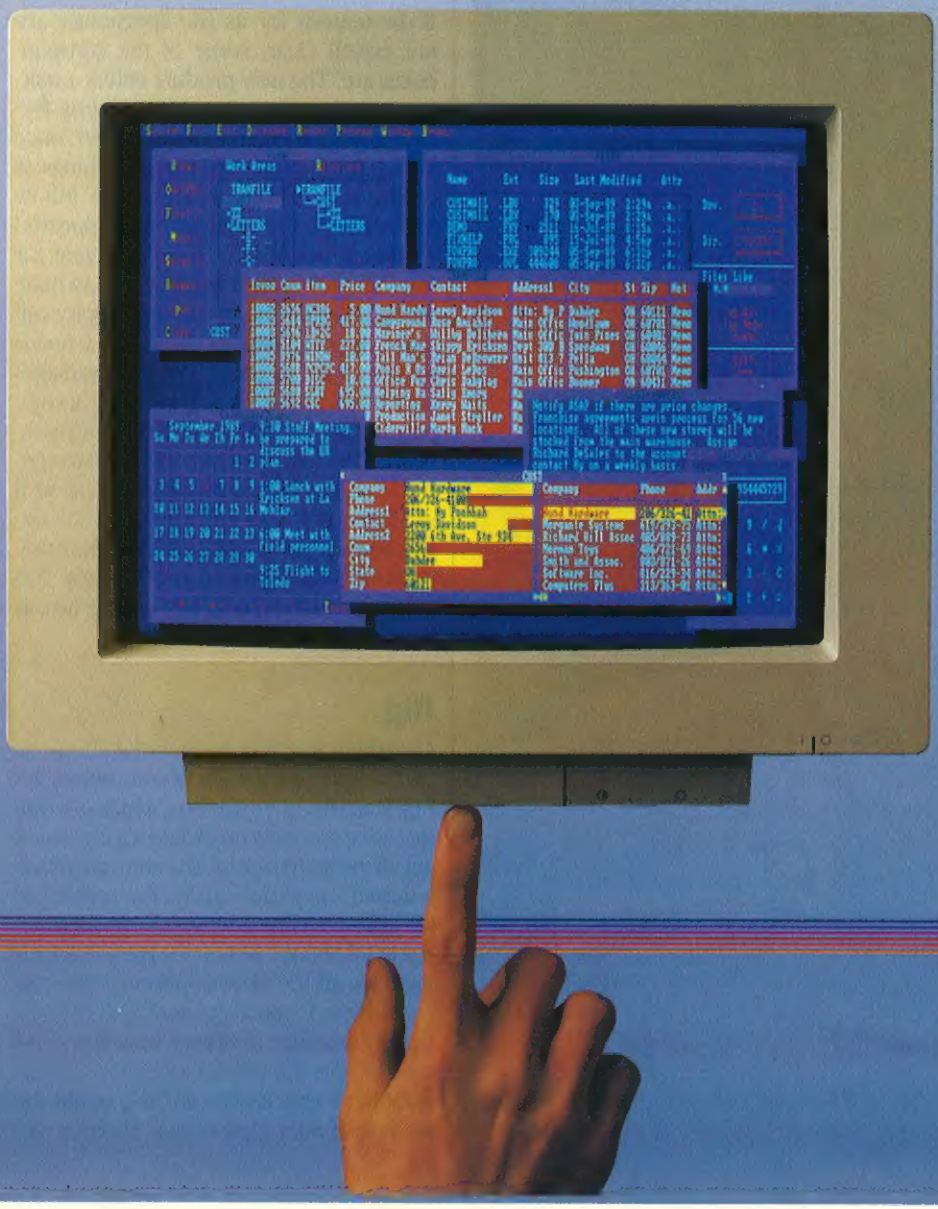
FoxPro is up to eight times faster than dBASE IV—more than 15 times faster than dBASE III PLUS!

And that blazing speed translates into unprecedented power. Now you can efficiently process gigantic databases with hundreds of thousands—even *millions*—of records.

Protecting Your Investment

With FoxPro, your existing FoxBASE+ or dBASE III PLUS programs will run perfectly—first time, every time, no excuses. And FoxPro is language-compatible with dBASE IV. But FoxPro doesn't stop there. It has over 140 language enhancements not found in any version of dBASE. We've outdone ourselves by adding more than 200 language extensions you won't find in FoxBASE+.

Best of all, FoxPro opens up whole new worlds for your applications by letting you move them onto a variety of different platforms.



The Tradition Continues

Fox Software is committed to excellence—our products prove it.

We've been producing superb database management software since 1983. And our products for both the PC and the Macintosh continue to win awards worldwide.

We've taken everything we know about software engineering, databases and interface design, and focused it into one remarkable product—FoxPro.

FREE Demo Disk

Shift the balance of power in *your* favour by trying FoxPro for yourself.

Call (0462) 421999 now to get your free demo disk. Or ask for the FoxPro dealer nearest you. See for yourself: *Nothing Runs Like The Fox.*

FoxBASE+ Users: Call About Our Upgrade Offer!

System Requirements: *FoxPro operates in 512K RAM (640K recommended) with MS/PC-DOS 2.0 or greater and an 8086/8088, 80286 or 80386 microprocessor. For optimum performance, FoxPro takes complete advantage of any available EMS (expanded memory) or a math coprocessor.*

Trademark/Owner: FoxPro, FoxBASE+/Fox Software; dBASE III PLUS, dBASE IV/ Ashton-Tate.

Fox Software

Nothing Runs Like The Fox.

Fox Software International
Intech House, Cam Centre, Wilbury Way
Hitchin, Herts, SG4 OAP, England
Tel: 0462-421999, Fax: 0462-421318

CIRCLE NO. 742

Lattice Come Home

Lattice Inc has released V6.0 of its C compiler, claiming an impressive set of added bells and whistles. We gave the software to Will Watts for some experimental ringing and blowing.

The Lattice C Development System Version 6 for DOS and OS/2 – henceforth 'Lattice C V6.0' – is one of the most long awaited upgrades in the industry. The previous major version release (V3) of this package came out in 1985; Lattice C V6.0 is on a par with dBASE IV itself as a a software tool not worth holding your breath for.

Why did Lattice delay so long in upgrading? Over a period of years, Lattice C has declined from being the market leader (such was its status that when Microsoft first began supplying a C compiler, it used a rebadged version of the Lattice's) to the more recent position, where surely only the faithful, the skinflint and those with vast amounts of Lattice C code to maintain still used it. Lattice blames the interval on the emerging ANSI standard ('we were unable to release Version 4 in 1986 because the ANSI committee took a little longer than expected'), OS/2 ('we realised that the USA market preferred OS/2 support to be bundled with the DOS compiler') and everything except Lattice itself.

If the reasons for its late appearance are not crystal clear, some of the consequences are. The new product enters a market with the reputation of its elderly forbear a hindrance, not a help. Where once Lattice set the standard for such things as calling conventions, it must now follow Microsoft. If the upgrade had appeared earlier, it would have been sufficient for the compiler to have been tweaked a little: an optimisation here, an extra library call there. As things stand, Lattice C V6.0 needs to demonstrate considerable advantages over its competitors. The publicity accompanying the product's release in effect acknowledges the point, comparing it to specialist super- optimiser Watcom C as well as Microsoft C V5.1 and Turbo C V2.0. Lattice found that its software beat the competitors into the ground at every turn. This was not, however, quite my experience, as you shall see.

Big

One thing that may be asserted about the package without danger of controversy: it's big. It arrives in a huge box, which is so full that once you have unpacked it, you cannot get all the stuff back in. The software installed itself straightforwardly. The twelve 360 KB disks contained enough archived material to fill about 7 MB on my hard disk (this includes all the various optional libraries and utilities). Lattice creates (to my eyes) a slightly peculiar directory structure: `include` files are placed together with the compiler's executables in \LC, while the memory-model dependent libraries are

POWER AND FLEXIBILITY

BRIEF 3.0 increases your productivity for just £149.95*

*VAT extra, FREE delivery

BRIEF has established itself as the leading programme editor on the market today. Now, with the launch of the latest enhanced version – BRIEF 3.0 – it offers even better productivity.

ENORMOUS UNDO POWER

BRIEF 3.0's enhanced UNDO capability takes the fear out of programming. BRIEF 3.0 allows you to undo any command that affects the cursor or the text (including global changes) up to 300 times, even back to the beginning of your session. It's the safety net you need to program with confidence.

REAL FLEXIBILITY

There's built-in support for 36 languages including Assembler, BASIC, C, COBOL, dBase, FORTRAN and Pascal. And thanks to its complete macro language you can make BRIEF 3.0 fit your own programming style with ease: change key assignment, commands, prompts, colours – even the indenting styles.

BRIEF 3.0 – new benefits

- With our new C-like macro language, CBRIEF, C programmers can write macros on the fly and not have to switch between C and a different macro language
- The new source level debugger works with both the original BRIEF syntax and the new CBRIEF macro language
- Maximum lines per file has been increased to 4 billion
- We've added to and improved template editing and smart indenting for ADA, Cobol, Basic, Fortran, Modula-2, Pascal and C
- We've added editable, multiple keystroke macros that can be saved and restored
- You can even save and restore your window set-ups

Plus the features that made BRIEF famous

- Flexible windowing
- Unlimited number of files
- File size limited only by desk space
- Regular expression search and replace

No other editor puts this much power and flexibility in your hands. So if you're looking for extra productivity purchase your copy of BRIEF 3.0 today.

Solution Systems

SOLUTION SYSTEMS are the UK distributors for BRIEF 3.0. When you buy from us you get the authorised UK version of BRIEF 3.0 dual media (3½" and 5¼") plus bonus macro diskette.

Special phone line for Access/Visa card holders. Save time by phoning 0763 73457 between 9.00 and 6.00 Monday to Friday to place your order.

N.B. Official purchase orders from local authorities, PLCs and educational establishments are accepted by post or by Fax.
Fax no: 0763 73460

BRIEF 3.0 The Programmer's Editor

Send to: Solution Systems UK, The Red House, 84 High Street, Buntingford, Hertfordshire SG9 9AJ

Please rush me my copy of BRIEF 3.0

☐ **I enclose cheque to the value of £172.44 (£149.95+VAT) crossed and made payable to Solution Systems UK.**

Orders despatched same day.

NAME ADDRESS

POSTCODE

☐ **Please charge my Access/Visa* account the sum of £172.44 (£149.95+VAT)**

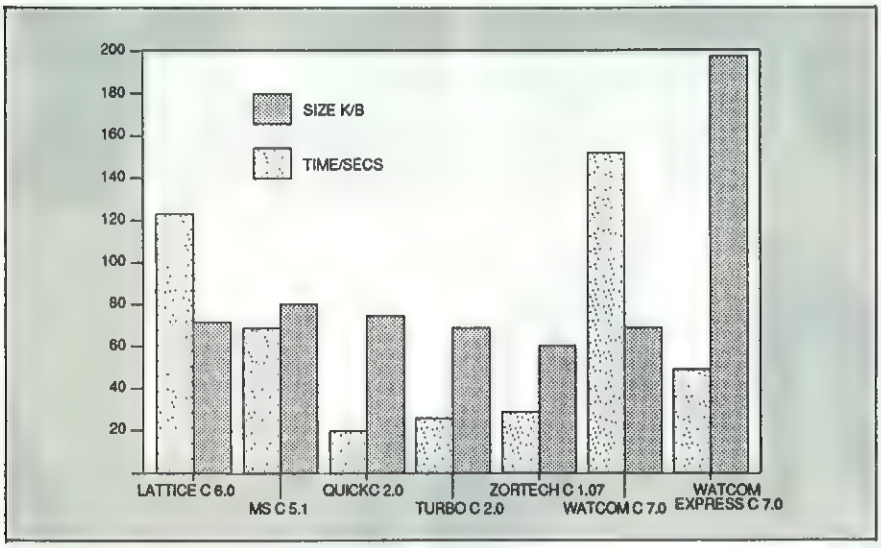
Card No **Expiry Date** **Signature**

*Delete as appropriate

.EXE 2/90

CIRCLE NO. 743

Figure 1 – Compile/link times and .EXE file size of test program



placed in separate directories of their own; \LC\S for the small model, \LC\L for the large model and so on. Incidentally, Lattice C V6.0 supports five memory models: small, medium (which Lattice calls 'large program'), compact ('large data'), large and huge. There is no support for the tiny model; presumably the company felt that it was beneath the dignity of an OS/2 compiler to produce .COM files.

The compiler itself is invoked from the C: prompt with the LC command. I use the term 'C: prompt' as opposed to 'MS-DOS prompt' advisedly: nearly all the Lattice utilities are fully- fledged family applications and are able to run in both real and protected modes. The exceptions, the editor and the debugger, are supplied in duplicate and differ very little in their operation. LC is similar to Microsoft's CL program, but is more versatile: as well as orchestrating the compiler phases and Lattice's linker, this program can also invoke the assembler, librarian and binding (family application production) tools. Naturally, the LC command has dozens of option switches, many of which are described in the documentation (but see below for a more detailed whinge about the manuals).

How does the compiler's performance compare with its peers? The chart in Figure 1 shows the results from compiling a 5300 line program with Lattice C V6.0 and six rivals, with default options (ie no optimisations). From this you may gather that Lattice C V6.0 produces files of about average size, but lumbers a little to produce them. This is all right by me, especially as the parser is very good at throwing up LINT-like warning messages (it caught me indexing into a two dimensional array with an outsized constant; something you expect effete Pascal compilers to pick up, but not macho, devil-may-care C), but it might be slow for the programmer who uses it in a 'twiddle the code, then try it and see' fashion. In the area of optimisation, however, things are different.

Look at the object code produced by the compiler for optimisation Test 1 (shown in Figure 2). The perfect implementation of the pointless function `testfn()` – the single instruction 'RET' – has eluded the program, but by gum! it was a close run thing. Lattice is the first compiler which I have seen that can decide to implement functions 'inline', like macro calls. Implementing `dummyfn()` as an inline function led the optimiser to the discovery that the call had no global side effects and could be eliminated. This places Lattice easily at the top of the league for Test 1 and a very close second to Watcom V7.0 for Test 2, as you can see in Figure 3. (Careful readers of previous articles will have spotted that I am guilty of revisionism. Watcom's speed for Test 2 has improved by 58 cycles – I'm afraid that I previously allotted five cycles to an instruction which only takes three. Sorry.) A clear points win to Lattice, then; and I must develop some more challenging tests to give the next generation of optimisers something to bite on!

How about ANSIfication? The accompanying blurb bills this as the ANSIest compiler yet. It probably isn't bad – I defer to BSI expert and .EXE writer Neil Martin in these matters – but I must report that the *very first program* that I created under Lattice failed to compile with Turbo C because of a lack of a Lattice ANSI violation (NULL not defined in STDDEF.H, see Section 4.1.5 of your Draft Standard). The ANSI-conformant compiler remains unwritten.

Lattice's most important first with this package is to do with the way the dual operating systems are handled. Suddenly, producing code for OS/2 seems natural, almost. The compiler can generate real/family/protected mode programs irrespective of the environment from which it is invoked; so one could use it to develop and debug OS/2 family applications without actually having a copy of the operating system. The package includes everything you need to create OS/2 V1.0 applications and DLLs: compiler, LINK/2 compatible linker, header files and libraries. To develop for OS/2 PM, however, you will need a copy of Microsoft's PM toolkit. Lattice does not claim to support Microsoft Windows, by the way, but thought that the compiler's code 'might well work'. The problem area, according to a Lattice representative, is the reliance of some versions of Windows on special prologues being generated at function entry points. Microsoft has not published the details of these prologues.

A Private Function

Lattice has introduced some non-ANSI keywords to cope with problems specific to the OS/2 environment. A `private` function is one that sets up its own data segment on entry. This is useful when writing functions for use in DLLs. Since any particular function may be called simultaneously in different threads, it needs to maintain several copies of its static data to achieve re-entrancy – which is the facility that `private` keyword provides. Global data may – rather confusingly – also be declared `private`. This causes a fresh copy of the data to be duplicated for each new thread.

The `pad` keyword, and its counterpart `nopad`, are used to control the length of structures. To explain: Lattice has chosen not to support huge (>64 KB) objects whose components straddle segment boundaries. Thus given the declaration

```
struct FOO
{
  int x;
  float y;
}; /* 6 bytes long */

struct FOO far stuff[100000];
```

the element `stuff [10922]` straddles a segment boundary. The solution is to pad out `FOO` to a length which is a power of 2 – in this case to a length of eight bytes. The `pad` keyword does this automatically, but it's up to you not to create structures which require (16 KB + 1 byte) storage.

That Documentation

Just thinking about the documentation makes me cross. Thanks to the crafty omis-

BORLAND INTRODUCES TURBO PASCAL 5.5 WITH OBJECTS

Be Objective.

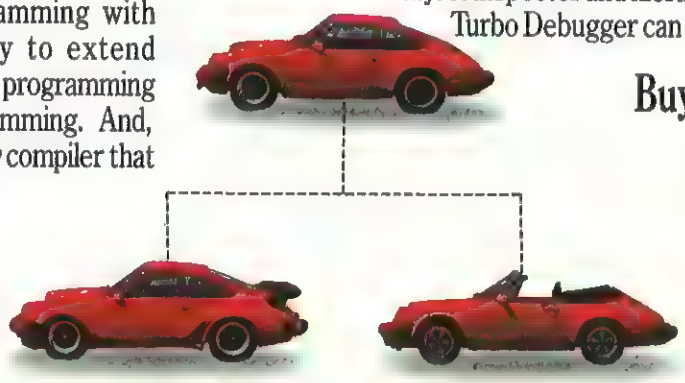
Turbo Pascal,® the world-standard Pascal compiler, adds Object-oriented Programming with our new version 5.5. We combined the simplicity of Apple's Object Pascal language with the power and efficiency of C++ to create Turbo Pascal 5.5, the object-oriented programming language for the rest of us.

It's easy to extend yourself

If you're already programming with Turbo Pascal, it's easy to extend yourself from structured programming to object-oriented programming. And, Turbo Pascal 5.5 is the *only* compiler that is 100% source-code compatible with your existing Turbo Pascal 4.0 and 5.0 programs.

A fast object lesson

Object-oriented application programs more closely model the way you think. Objects contain both data and code. As in a spreadsheet cell, the value and the formula are together. Objects can *inherit* properties from other objects. For example, a Porsche Carrera inherits most attributes from base model 911, but it also sports a whale tail.



Inheritance provides powerful modeling capabilities by allowing objects to inherit attributes from other objects.

Turbo Pascal 5.5's object-oriented extensions give you code that's easier to change, extend, and support.

Turbo Pascal 5.5 Professional with Turbo Debugger® and Turbo Assembler®

The award-winning Turbo Debugger now includes an object inspector and hierarchy browser. And Turbo Debugger can debug any size program.

Buy objectively

Turbo Pascal 5.5 costs £99.95 plus VAT and Turbo Pascal 5.5 Professional, including Turbo Debugger and Turbo Assembler costs £199.95 plus VAT.

For further information, simply talk to your dealer or call our PRODUCT INFOLINE for a copy of our Turbo Pascal 5.5 brochure on (0628) 771070.

Turbo Pascal 5.5 Features

■ Inheritance ■ Static & dynamic objects ■ Constructors & Destructors ■ Object constants ■ Compiles @ > 34,000 lines/minute ■ New integrated environment tutorial ■ Hypertext Help with copy and paste ■ Enhanced smart linker & overlay manager ■ Support for 8087/80287/80387 ■ Integrated source-level debugging

B O R L A N D

Borland International (UK) Ltd.,
8 Pavilions, Ruscombe Business Park
Twyford, Berkshire RG10 9NN

CIRCLE NO. 744

Figure 2 – The optimisation tests

Test 1 - C source code

```
/* Test 1 */

void testfn(void)
{
  int e = 0, f = 0; /* e and f simply used as */
  int i, j, k;      /* constants i, j and k are all */
  i = 7;            /* redundant & may be eliminated */
  j = 3;
  i = i + 3 + 8;
  j = 2 + i + j + 9;
  dummyfn(e,f);     /* Must make this function call...*/
  i = (i + 3)/2 + 8/3;
  i = -4;
  i = (i + 3)/2;
  j = 2 + i + (j + 9);
  dummyfn(e,f);     /* ...and this one */
} /* testfn */

int dummyfn(int param1, int param2)
{
  return param1 + param2;
} /* dummyfn */
```

Test 1 - Lattice C object code

```
;Test 1: Lattice C Compiler V6.01
;Takes 21 instruction cycles to run
;(minimum possible - a 'ret' instruction - takes 11)
;
;'dummyfn', which is not called, would take
;36 instruction cycles

testfn      PROC NEAR
    push    bp          ;makes stack frame which it
    mov     bp,sp       ;doesn't need
                        ;compiler has turned dummyfn
                        ;into an inline function,
                        ;realised it doesn't do
                        ;anything useful, and
                        ;optimised it out.
    leave               ;dispose of stack frame
                        ;that we didn't use
    ret
testfn      ENDP

dummyfn     PROC NEAR
    push    bp
    mov     bp,sp
    push    cs          ;why does Lattice use a
                        ;push instruction to
                        ;reserve extra stack?
    mov     ax,[bp+4]
    add     ax,[bp+6]
    leave
    ret
dummyfn     ENDP
```

Test 2 - C source code

```
/* Test 2 */

/* Assume these (local) declarations */
int i,j,k;
int array[30][20];

  |          |
  |          |

/* Results show translation of this loop */
for (k = 0; k < 30; k++)
{
  array[i][j] = array[i][j] + array[i][j+2] + array[i][j+1] + k;
}
```

Test 2 - Lattice C object code

```
;Test 2: Lattice C Compiler V6.01
;Takes 1170 instruction cycles to run

                        ;ax contains i
                        ;di contains j
    xor     si,si       ;k goes in si
    mov     bx,di       ;pointless 'mov'
    shl     bx,1        ;could have used di
    imul    cx,ax,40    ;get i * row width
    lea     ax,array    ;mistake #2 - could do
    add     cx,ax       ;this in loop without
                        ;(286) time penalty
    mov     di,cx       ;another pointless reg switch
    add     di,bx       ;ready to go now
L1:
    mov     ax,[di]     ;get array[i][j]
    add     ax,[di+4]   ;add the other components
    add     ax,[di+2]
    add     ax,si
    mov     [di],ax     ;store safely
    inc     si          ;this is k
    cmp     si,30
    jl      L1          ;end of loop?
                        ;all done
```

sion of a description of the optimisation switches, the compiler nearly came fourth instead of first in the fastest code competition. The day was only saved by a very late, chance re-examination of the READ.ME file. But this is not the only stimulus to my ire.

The manuals, I should explain, take the form of four IBM-style ring binders. These are all filled to the point where the rings spring open and disgorge the contents at the least provocation. Volume 1 covers the compiler operation and utilities, Volumes 2-3 the libraries and Volume 4 contains the assembler's material and a giant index. It is certainly impressive in size, but, as we all know, size isn't everything.

In its favour: it's not hypertext, the text is clear and easy to understand and the print is good quality. The debit side of the ledger dominates: information is difficult to find, left out entirely or just wrong. Here are some more examples to back up my case. Pre-defined macro symbols, such as `_ANSI` and `LATTICE`, were explained only in the compiler switches section and do not feature in any indexes. Descriptions of the interrupt enable/disable functions `CLI()` and `STI()` were, like the optimiser switches, completely absent – one begins to suspect that the program is really a C++ compiler, but they just forgot to say so in the manual. The code fragment given to illustrate the use of the `ComSendK()` function shows the wrong number of arguments, ie no one bothered to test-compile the examples before putting them in the manual.

A Man from Lattice tells me that the documentation is currently being revised, with a re-issue scheduled for March. If you have nothing better to do with your time than thrash through hundreds of pages searching for non-existent references, don't bother to wait.

The Libraries

A major selling point for Lattice C V6.0 is the extensive bundled collection of libraries. As well as a complete set of ANSI functions – including the obscure recent ones like `setlocale()` – there is a full selection of UNIX functions, mathematics functions (hands up everybody who used `yl()` – 'first order bessel function of the second kind' – in the last six months), specialised protected mode calls for thread handling, and the usual range of de facto standard functions: things like `kbhit()` for monitoring the keyboard buffer and `cprintf()` for printing directly to the console. There is one snaglet. Lattice C uses a different set of header files from Microsoft/Borland/Zortech/Watcom. This seems silly: C code that uses only these plain vanilla calls should work without change. It would surely be to Lattice's advantage to make sure this happened.

Missing from this base library are those low-level functions, with names like `setvect()` and `keep()`, that let you put together TSRs without dirtying your soft, clean C programmer's hands with assembly language. Lattice says that this omission is caused by them not yet having got around to it (rather than deliberate policy) and that an update will follow in due course.

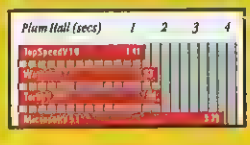
As well as the core set of system functions there are four application libraries. The

Can your compiler meet the challenge?

We invite you to take an existing program and compile it using TopSpeed C. Then, compare the overall performance with the compiler you now use. If you are not 100% satisfied, return the entire package to us, and we will refund all of your money.

	TopSpeed* C Version 1.00	Microsoft* C Version 5.1	Turbo C* Version 2.0	Watcom* C Version 7.0
100% ANSI compatible*	✓			
Integrated environment	✓		✓	
Pass parameters in registers	✓			✓
Expand any function as inline code	✓			
Supports OS/2	✓	✓		
DOS Dynamic Link Libraries *(overlay code linked at load time)*	✓			
Smart linking *(only referenced code and data linked into .EXE)*	✓			
Type-safe linking *(function parameters and memory model checked at link-time)*	✓			
Fully automatic make works across libraries	✓			
Time-sliced scheduler for multi-tasking under DOS	✓			
Short pointers in any segment	✓			
Hypertext help with library online	✓		½	

*Written by Ned Martin of the British Standards Institution (BSI) and printed in Personal Computer World June 1989, page 241.



Standard Edition—
Optimizing DOS compiler, integrated source-level debugger, smart linker, automatic make, complete ANSI library, 6 memory models, BGI interface, time-sliced scheduler, MS-DOS/BIOS interface, superset of MS and Turbo C libraries, and more

Extended Edition—
Standard Edition plus full source code to libraries, post-mortem debugging, MS Windows support, support for DOS DLLs (overlays), assembler disassembler, profiler, DOS call monitor, and more

OS/2 Edition—
Extended Edition (except for DOS compiler) plus ability to generate DOS executables, full support of OS/2 Kernel & PM calls, OS/2 smart linker fully automatic generation of DLLs, source for protected mode libraries, and more

TopSpeed C:

Standard Edition £149.00
(DOS compiler and VID)
Extended Edition £295.00
OS/2 Edition £370.00

All prices are excluding VAT, postage and packing

64 page TopSpeed C Technical Specification available on request

Phone for TopSpeed Modula 2 compiler and toolkits technical information and prices

To Order:

Phone: 01 253 4333
Fax: 01 251 1442

Phone for full details

Credit cards Visa/MC Access

Jensen & Partners UK Ltd
Unit 21
63, Clerkenwell Road
London EC1M 5NP

30 days unconditional money back guarantee
30-day unconditional money-back guarantee

To buy in the USA and Canada Jensen & Partners International
1101 San Antonio Road
Suite 301 Mountain View
CA 94043
Phone 415 967 3200

Standard Edition	$199
Extended Edition	$395
OS/2 Edition	$495

1101 San Antonio Road, Ste. 301
Mountain View, CA 94043
Phone: (415) 967-3200
Fax: (415) 967-3288

TopSpeed®

TopSpeed and TechKit are trademarks of Jensen & Partners International. Other brand and product names are trademarks or registered trademarks of their respective holders.

CIRCLE NO. 745

Figure 3 – Optimisation tests

	Test 1	Test 2
Lattice C	21	1170
Microsoft C 5.1	221	1407
QuickC 2.0	304	6126
Turbo C 2.0	261	6773
Zortech C 1.07	127	1715
Watcom C 7.0	66	1161
Express C 7.0	728	19454

graphics library suffers from the fact that it can only be used in real mode because, as the documentation quips, 'it is more natural to use the Presentation Manager graphics facility when designing for OS/2'. There are 44 functions, which provide you with all the functions you need to get things done, except for perhaps for text/font manipulation (the functions use the ROM bit-mapped fonts). I have been using this library heavily and have found the code reliable and fast. All the standard graphics card types from Hercules to VGA are supported. There are no routines for 3D histograms or pie- charts.

The Curses library is an emulation of the UNIX System V screen manipulation interface. The Curses is based around simple text- based windows which are first put together in buffers, then flushed to the terminal or screen. The library works in both real and protected modes, so it may be used to create applications portable between MS-DOS, OS/2 and UNIX. The dBC III library lets you create, write and read from dBASE III PLUS compatible .DBF, .NDX and .DBT files. It also runs in both modes – a suggested use is a PM application that accesses a dBASE database – as does the final library, the Communications Library. This contains a set of routines to drive the serial port. The functions range from `ComGetc()`, which returns the next available character from the buffer, to `ComSendK()`, which transmits a complete disk file in accordance with the Kermit error-checking protocol. In the version of the compiler that I tested (V6.01) this function, plus its XModem and YModem brothers, didn't work properly. The rate of transmission was sometimes very slow (for a given baud rate), files were invariably garbled and the programs crashed frequently for not much reason. Lattice tells me all this known about and is to be fixed in later versions. Even if it this were the case I should still have doubts about these functions – they do not give the programmer sufficient control over important things like errors, and where the incoming file should go.

Utilities

There are many utilities, of varying quality, supplied with the package. LSE is a full screen editor. You can invoke the compiler from within it; it can point out the lines where syntax errors occurred. It boasts a number of moderately neat features to aid the writing of C and assembly language programs; for example, it indents eight spaces after a line ending in a {. It has a configuration program, for setting up your own key mappings, which often works. Its keystrokes are not based on any other editor that I have met. I confidently predict that, if you are already familiar with any other full screen editor, you will leave this one well alone. It's not that it's especially bad; it just doesn't offer enough to tempt you to change.

The source level debugger is called CodeProbe. It is a very workmanlike piece of software. Unlike Watcom's offering WVIDEO, CodeProbe knows how to how save a graphics screen (sorry about my obsession with graphics modes; I have been working on a graphics program for some weeks). As is now the fashion, there is a 'remote debugging' option, where a kernel program on the target PC is connected by a serial link to a front end module running on a host. The idea is to maximise the amount of memory available to the test program. This lash-up did not work very well for me. The kernel program printed 'Press Ctrl-Break to end' but wouldn't when I did, and the front end kept crashing. However, the debugger worked fine on a single machine. There is little to choose between CodeView and CodeProbe as an OS/2 debugger. CodeProbe's debugging information, incidentally, is held in a proprietary format. You cannot use it to debug programs produced by other compilers.

The assembler, LASM, is really none of my business: one macro assembler seems very much like another. The documentation claims that it is very nearly MASM compatible, and that it can handle all iAPX opcodes up to 80386 native mode and 80387. The program correctly translated the fragments that I offered to it. There is also a simple make utility called LMK, a TOUCH program, a cross-referencer and, finally, GREP and various other little programs, reeking of UNIX, for doing terribly clever things with pipes and redirection.

Conclusion

Lattice needed to catch up a lot of ground with this product. To some extent the company can claim to have done this: the compiler has OS/2 support, a gargantuan set of library functions and a white-hot optimiser. And yet, and yet... What about the documentation? Will Lattice put it right at the second attempt? What about all the little bugs in the programs? Should one hang on until it stabilises? What about the new compilers rumoured to be imminent from Borland and Microsoft?

One thing at least is clear. I promised to buy a pint for the creator of the first compiler to out-optimise Watcom. I thought my beer would be safer for longer than four months. However, if Lattice Inc would care to send a representative to the Barley Mow pub, there's a bottle of Newcastle Brown waiting there for him. EXE

Lattice C V6.0 costs around £165 and is available from dealers.

Figure 4 – Screen from CodeProbe, the debugger

Letters

We welcome opinions on any subject that is relevant to software development, especially feedback from articles published in .EXE. Please write to The Editor at 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked Not For Publication it will be considered for inclusion on this page.

Dear .EXE,
We were very pleased to see our product, Protec, mentioned in your feature on data and software security in the November issue of .EXE. However, we were a little less pleased with your inaccurate comments regarding the product, especially the author's alleged break-in at a security conference.

First, the author of the article said that Protec does not trap Int 13h calls, and that consequently the hard disk was totally open to any software that uses Int 13h calls. As we all know, Int 13h is a ROM-based routine and it is impossible for any software to trap these calls, except using special features of the 386 in protected mode.

The author also stated that he had used Norton's Disk Doctor to crack the Protec security and unlock the hard disk. This is misleading. Using NDD trashed the disk's file allocation table (FAT), allowing a program that uses Int 13h to directly read sectors from the disk. This is not the same as disabling the security, in my opinion.

You mentioned that Clam, a competing product, protected itself from Int 13h programs by providing encryption, and you stated that Protec did not. This is not true. Protec also has an encryption facility as standard, but we had not enabled it because the conference organisers had told us not to. I do not understand why Clam's encryption was turned on.

Paul McKibbin
International Data Security

Robert Schifreen replies:
First, apologies for paraphrasing your letter, Paul. As you sent it to me, it would have filled over 2 pages in .EXE, and we simply didn't have the space. However, I hope that I have repeated all of your concerns. The security conference at which I used NDD to break Protec was indeed one organised by the IBM PC User Group. I was shown a desk, on which were 2 machines. One was protected by Protec, and the other by Clam. I was given no manuals, and no clues. NDD did, as you say, trash the disk's FAT, and this bypassed the initial protection. I could then boot from a floppy disk and use an Int 13h call to read the hard disk. I was not informed that Protec had an encryption feature and, because Clam's was clearly enabled, I assumed that Protec lacked such a facility. I'm sorry if I gave the wrong impression, but I believed at the time that the person who had set up Protec had installed all the options.

Although I do not dispute that you were asked not to install the encryption facility, I am very surprised that your representative, who was watching me throughout, did not raise this matter. As soon as I had bypassed the initial security, he sulked in a corner and gave the impression that he was admitting defeat. If, as you say, Protec had sufficient extra features to have kept me out, I would like to have been told.

I understand that a few users have tried Norton Disk Doctor on their own Protec-protected machines, and that you have had to engage in various circuitous routes to retrieve their FATs. I hope that publicising the situation in .EXE, as your letter has done, will prevent this happening again.

Dear .EXE,
I read with interest the article in December's issue, entitled 'The Thinking Programmer's Guide to UARTS', having recently been involved in programming for real-time data capture via the PC serial port. There is, however, one point in the article which contradicts the information that I have on the 8250 chip.

In Figure 5, a note at the bottom states that National Semiconductor recommend that the Line Status Register is not written to. However, one of the advertised features of the 8250 is the ability to read and write all the control and status registers. My Nat Semi datasheet for the 8250, in referring to the Data Ready bit (bit 0 of the Line Status Register) actually says that bit 0 'may be reset to a logic 0 either by the CPU reading the data in the receive buffer, or by writing a logic zero into it'. The ability to do this can be very useful.

It is quite possible for the receive buffer register to contain an unwanted character, when a peripheral has been connected or powered up. As soon as an interrupt routine is established and enabled, this character will cause an instant interrupt and will be read by the routine. To prevent this, it is simpler to invalidate the character by resetting the Data Ready bit in the Line Status Register, rather than read and discard the character. I do this in my own programs, and have had no problems so far.

John Keutgen
London NW10

Dear Sir,
I feel it would be beneficial if I clarify 2 points that were made in the December issue of .EXE. First, SideKick for Presentation Manager will run under OS/2 1.1, and not solely under 1.2 as your news story suggested. The use of the new facilities in OS/2 1.2 will not affect downward compatibility.

Second, in regard to your letter about Turbo Basic from Mr Duffin. Owing to the dynamic nature of strings in Turbo Basic, a descriptor must be maintained for each string. For faster reference, this descriptor is stored within the data segment, thus

VARSEG and VARPTR will refer to the data segment and not the string segment. I have enclosed a short routine which extracts the correct location of the string segment from the data segment, which will show Mr Duffin how to do what he was trying to achieve.

Matthew Price
Senior Technical Support Analyst
Borland International, (UK) Ltd.

Robert Schifreen replies:
We have Matthew's short BASIC program on file. Anyone who would like a copy can call the .EXE office, send an SAE or leave us a Fax number.

Dear .EXE,
In your December issue, you ask for an 'accurate and succinct' definition of the term 'real time'. I think you have fallen into a trap which is all too often encountered by people in this industry. Because we are used to dealing with the precise and deterministic nature of computers, we expect that the rest of the world should exhibit these characteristics. I feel that your question is rather like asking for an accurate and succinct definition of the colour green!

Of course we can all come up with a *description* of the colour green, eg somewhere between yellow and blue in the visible spectrum, but this is not a definition.

So, do I have a *description* of real time? Yes, of course. It depends on the concept of the *value* of information generated by the system. In most cases, this value tends to fall as time progresses, ie as the information becomes older and staler. If this value could be quantified, then in some cases we would see a fairly linear fall, while in others we might see something which started off fairly flat but, at some point, started to decline rapidly.

Three examples come to mind. First, a system which is generating a natural language dictionary. The value of this system's output does not decline over the time it takes to produce the document, regardless of whether the production process lasts an hour or a week.

Second, a system which is monitoring a patient's respiratory rate. This very suddenly loses value if it cannot detect and notify a significant pause quickly enough.

Finally, a weather forecasting system. This is useless if it cannot produce a forecast for the next day within 24 hours, and it is progressively more valuable as that time decreases.

So which of these are real time systems? I think that the answer to that question depends not only on the application, but also on the supporting technology, so we describe a real time system as one where it is technically difficult to achieve the system's outputs within the time period where those outputs retain their value, or retain sufficient of their value to warrant using the system. Of course, this line of thought means that what is a real time system today may not be one tomorrow, assuming that the system requirements stay the same. Weather forecasting systems of yesteryear might, by this description, have been real time systems in their day but, if the requirements for accuracy and reliability had not changed, they would not be so considered today.

Bernard Northmore
Science Systems Ltd
Bristol

Dear .EXE
I write in response to a letter in November's issue from Linda Allen, regarding the ability of the ANSI.SYS driver to redefine keys on a PC.

I produce commercial software that not only uses ANSI, but does (on occasion) redefine keys. The use of colour, cursor movement and general screen control is much easier with ANSI, than having to use BIOS calls.

I admit that it is possible for a practical joker, and the more sadistic in the profession, to redefine keys using ANSI so that they do much more nasty things. I myself have been on the receiving end of such pranks. However, since I work in an environment dealing with a lot of users' software (sent as backups for diagnostics), which I have to restore to my PC, there are several precautions that I take.

The first is to test for viruses using IBM's virus detection software, plus a few other utilities that I have collected. The second is to look for filenames that I don't recognise. The major fail-safe, of course, and the one that we should all take, is to keep regular backups.

You can, with a language like C or Pascal, write a little program that looks for ANSI escape sequences and gives you a list of files to avoid if you can't risk it, or look at with an editor. The point is, though, that I rarely find anything that is either malicious or even just plain daft. Vendors of respectable software will not do nasty things to redefine keys, although errors may occur that do end up fatal.

The onus is on the user not to acquire 'dodgy' software and, if they must, to take precautions.

Greg Lomax
Kewill-Xetal Systems Ltd
Salford

Robert Schifreen replies:
Of course, as you say, it is up to users to keep regular backups. However, the majority of users do not do this, which explains why a handful of people in the UK can make a fairly good living by recovering data from disks which should have been backed up in the first place.

Unfortunately, the number of ways of acquiring 'dodgy' software are numerous. There are tens of thousands of PC programs on bulletin boards around the world, all of which are available for the price of a phone call. Thankfully, the current virus scare has not stopped the writing and distribution of these very useful pieces of software.

My own feelings toward ANSI.SYS are that it is not only dangerous (for reasons already discussed), but extremely slow. Performing screen control via DOS is sluggish enough, and ANSI only makes it worse. Of course, there are many cases where speed is not important but, even then, answering support calls from users who have forgotten to install ANSI.SYS and whose screens are consequently filled with left-pointing arrows and square brackets is no fun.

If I was forced to use a program which performed its screen control with the ANSI driver, I would set up a batch file which loaded ANSI.SYS dynamically before running the program (there's a program to do this on the .EXE Disk volume 2), and rebooted the machine afterwards. If nothing else, it saves a few K of precious RAM.

Dear .EXE,
I've seen programs that are distributed as bootable disks, and I'd like to distribute my software in this form. For a start, there are no installation problems (because there's no installation!). Also, with no operating system on the disk, there's a full 360K (or 1.2 MB) of space for the program and its data. Is there an easy way to do this?

John Pricer
Watford

Robert Schifreen replies:
There's a program on volume 2 of the .EXE Disks called NODOS. This will create a bootable copy of a program for you, on a 360K or 1.2 MB floppy. There is no operating system on the disk (hence no licence fees payable to Microsoft), but this means that your program cannot make any DOS calls. It can, of course, call the ROM BIOS. The program's written by Giles Todd, who also wrote the dynamic device driver loader, and it's available for instant downloading from the 'exe' conference on Cix, if you can't wait for the postman.

Dear Robert,
I have the need to insert characters into the keyboard stream of a PC, under the control of a hardware interrupt handler. This is to supply keystrokes to an application package from external hardware or from the actual keyboard in parallel.

Currently, the characters (plus scan codes) are placed into the buffer directly by my interrupt routine. This works fine, except with programs like SideKick that grab interrupt 9, the hardware keyboard interrupt, for some of their input. For example, my TSR is unable to pop-up SideKick, because it is SideKick's own Int 9 handler that looks for the activation code, and my TSR never sees it.

Can I load a scan code directly into the hardware register on the keyboard that Int 9 reads? I seem to remember reading somewhere that the AT-style keyboard controller is programmable – could it be instructed to send a specified scan code, and do you know where details of such information may be found?
Many thanks,

John Pote
The Pulse Business Center
Walsall

Robert Schifreen replies:
Int 9 is the lowest level at which you can intercept the PC's keyboard operation. The AT keyboard is programmable, as you say, but only to the extent of being able to change the repeat rate for when keys are held down, and the time delay after which keys actually start repeating. I can see no way of actually putting characters into the keyboard's hardware.
Have you experimented with the order in which you load your interrupt handler? Loading it before Sidekick may make a difference. Otherwise, it may be that your only solution is a hardware one. The keyboard connector on a PC is a (non-standard) 1200 baud serial port. Perhaps you could resort to squirting characters down there. Does anyone else have a less messy solution?

Dear Robert,
Well done! Your editorial in November's issue (how to listen to cellphone calls) affirms the validity of the defence to that infamous Prestel hacking charge of a few years ago. It also reminds me of the original Wireless Telegraphy Act which, recognising that it is impossible to stop people listening in to radio, had to be content with forbidding the use of any information thereby gained (which is another impossible thing to monitor in all but a few cases).

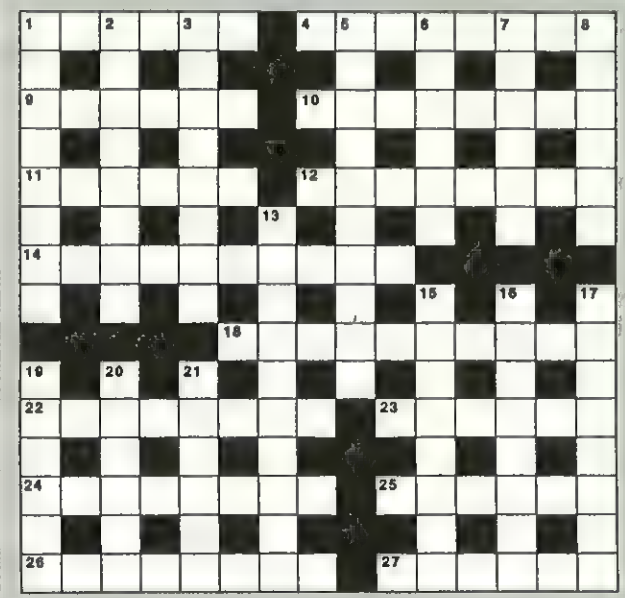
Those who urge the banning of eavesdropping (probably the best term to describe the deliberate, or accidental, act of listening in) sometimes liken it to opening and reading someone else's mail. I would describe it as a natural and harmless activity such as opening and reading a book which has been left lying around.

The best defence against security leaks is to assume that you have no defence.

David Tait
Dorking

Robert Schifreen replies:
My editorial regarding the ease with which it is possible to listen in to cellphone calls generated more response than almost anything else I have written for .EXE during my 2 years here. A number of large companies, including one of the clearing banks, called to ask permission to hand round photocopies of the article to all their mobile staff. If just one or two people have taken notice of what I wrote, and stopped giving out super-user passwords over the air, then I will have achieved what I set out to do.

.EXEWORD NO.3



ACROSS

1 I'm in a do about a checking system (6)
4 Trivalent impurity (8)
9 Hardware belonging to us strictly speaking (6)
10 Output units like Guys (8)
11 Noted in a 1-D array when off balance ... (6)
12 ... and fussy as the program did to the data (8)
14 Simple like the items in a structure (10)
18 Hopeful output of a maximum efficiency program? (10)
22 Exhibits screens (8)
23 Without a key, take a lan to ... (6)
24 ...Test thoroughly what a variable holds (8)
25 Vote against the final version of the program (6)
26 What GEM and such replace (8)
27 Made a home inside several loops (6)

DOWN

1 Processing offered by transputers (8)
2 Take account of the most central store (8)
3 Made the code more efficient? (8)
5 Home of silicon valley (10)
6 Admission that sent about two orientals (6)
7 Output of the elements of a tree I hear (6)
8 Stay in store about the team (6)
13 How to pass through a loop of refinery (4,3,4)
15 Bits able to become a flip flop (8)
16 Least productive part of the road where you may sit on set (8)
17 Formed a product like a 3D artist (8)
19 Final element of a sum that keeps its value (6)
20 Tries to produce chunks of good text (6)
21 Show off negative beginning in apartment (6)

.EXEWORD NO.2

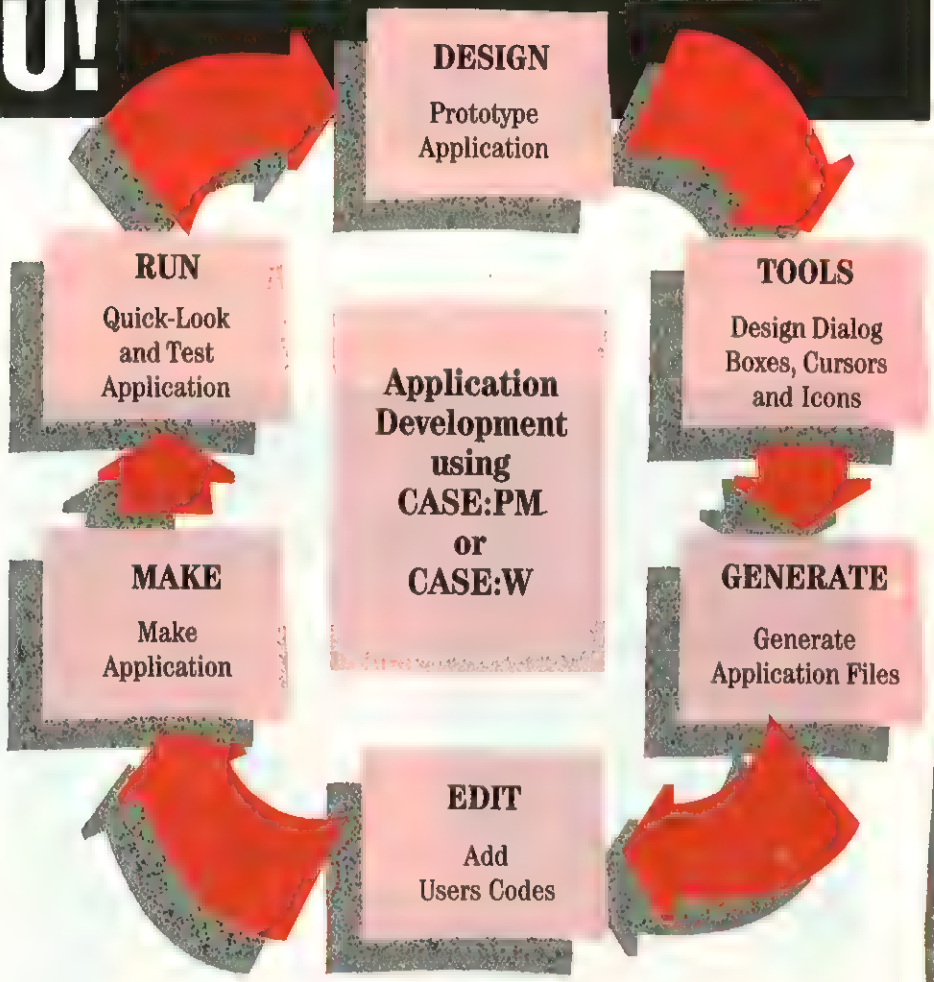
E	L	E	M	E	N	T	■	B	O	O	L	E	A	N
X	■	X	■	Q	■	H	■	R	■	N	■	R	■	O
E	X	C	L	U	S	I	V	E	■	E	P	R	O	M
C	■	E	■	A	■	G	■	A	■	B	■	H	■	A
U	N	L	A	T	C	H	■	K	E	Y	W	O	R	D
T	■	■	■	O	■	B	■	■	■	T	■	U	■	I
E	N	T	E	R	■	O	B	S	T	E	T	R	I	C
■	■	W	■	■	■	N	■	T	■	■	■	L	■	■
O	V	E	R	S	T	E	E	R	■	S	T	Y	L	E
P	■	N	■	C	■	■	■	U	■	E	■	■	■	X
E	X	T	R	E	M	E	■	C	L	E	A	N	U	P
R	■	I	■	P	■	T	■	T	■	K	■	A	■	E
A	D	E	P	T	■	H	E	U	R	I	S	T	I	C
T	■	T	■	I	■	Y	■	R	■	N	■	A	■	T
E	T	H	I	C	A	L	■	E	N	G	U	L	F	S

CASE:PM or CASE:W

THE TOOLS THAT WRITE PM OR WINDOWS CODE FOR YOU!

Writing applications for Windows or OS/2 Presentation Manager graphical user interface just became a lot easier, because now there's CASE:PM (for Presentation Manager) and CASE:W (for Windows).

CASE:PM™ Both are application prototypers and code generators that write complete C skeleton programs for you based on a description of the 'user interface'.



APPLICATION PROTOTYPING

Just by 'pointing and clicking' at various parts of a window you can now create a Windows or PM user interface – in minutes!

APPLICATION GENERATION

Once you have the 'look and feel' of the application, CASE:PM or CASE:W produces C source code, resource scripts, definition files and make files... *automatically.*

Run *MAKE* and the files are compiled and linked producing a standalone executable with no run-time environments or royalties to pay.

CASE:W™

APPLICATION MAINTENANCE

Once you have the skeleton you add the application specific code. The generated source code is fully documented, with instruction as to where to put the application specific lines, leaving you free to concentrate on the application itself.

Change the user interface design and CASE:PM or CASE:W will generate the source files with the new user interface – but retaining the code you've added.

CASE:PM or CASE:W will reduce your learning curve and increase your productivity. For information and technical specifications call the Products Group on 0285 655888 FAX (0285 650537), or write to us.

QA

Cecily Hill Castle
Cirencester, Gloucester GL7 2EF
0285 655888

CIRCLE NO. 747

FoxPro dBASE Without Tears

FoxPro is billed as 'what dBASE IV should have been'. Ian Turner tries it out.

Windows, windows everywhere, and not a dot prompt in sight. The horizons of the dBASE language have been expanded with the release of version 1 of FoxPro from Fox Software. FoxPro is, of course, a dBASE clone, but with this release it is perhaps difficult to decide with any certainty who is cloning who.

The interface uses text-based windows, pull down menus and dialogue boxes which is a refreshingly new and effective approach to the dBASE language. It makes extensive use of all three and, preferably, a mouse, although it will function with the keyboard alone. Despite the restrictions of using a text based screen rather than a true graphics screen, it follows as closely as possible the Apple Mac version of its sister product FoxBASE+.

FoxPro comes on two 360 KB 5.25 inch floppy disks, with a further three disks containing the FoxView and Foxdoc add-in utilities and sample programs. If required, 3.5 inch disks will be supplied. You create a suitable sub-directory on your hard disk, and then type A:INSTALL. You are prompted to change disks when needed. The initial installation requires you to enter the serial number and also an 'Activation code' from a slip of paper found in the now familiar sealed software envelope. You can enter a demonstration code rather than the full code, which means that you do not have to accept the terms of the standard software licence on the disk envelope to review the program.

Once FoxPro has been installed from the first two disks, the utilities are installed from within FoxPro itself, which is automatically loaded by the INSTALL program.

Message for frustrated C users: Give it up! Use PCL in DOS or OS/2

PCL will shorten your development times drastically. PCL is a better language with a richer function set and it is much easier to use and learn than C.

PCL does not crash your machine every five minutes while you are developing your program. You can display the contents of any variable or file, try out a PCL statement or call any procedure from the PCL command line.

So you don't have to study HEX dumps or look at registers to find out what went wrong. And you don't have to press Reset to bring your machine back to life.

PCL is an interpreter/compiler. It gives you the rapid, interactive program development only an interpreter can offer as well as the execution speed you expect from a compiler. Prototype and refine your application speedily and interactively. You can make PCL RAM-resident and use your favourite editor to hotkey from editing to testing with two key strokes.

PCL is a well structured language with sensible keywords. So you can write programs you'll still understand six months later. There are no reserved words, and no cryptic symbols. If you have struggled with C you'll appreciate that. And if you already know Pascal or Basic you'll learn PCL in a week.

PCL is used in 26 countries. With our special offer price of £55 for the single user DOS version you can afford to let your copy of C gather dust and switch to a useable language. If you want to exploit the power of OS/2, choose the new PCL OS/2 version.

Some key features of PCL :

- Sophisticated windows, boxes, frames and menues. Save and restore window coordinates as well as contents. Unlimited number of windows. Paint and redraw pop-up windows with ease.
- Extremely fast screen displays.
- Array arithmetic. Built-in sort for text and numeric arrays. Arrays can use all available memory (to DOS limit).
- Extended text functions for searching, translating, verifying, parsing etc.
- Decimal arithmetic and full scientific function set with 16 digits precision. DATE, HEX and BINARY arithmetic. Automatic 80x87 support.
- Powerful file handling and disk management. Switch between sequential and direct access at any time. Record and file locking for net-working. Fast ASCII and binary file modes.
- Dynamic record structures - fields can be added, deleted or reordered.
- User defined background tasking while programs wait for keyboard input.
- Built-in RS232 drivers for fully buffered interrupt driven communications at any baud rate from 2 - 19200.
- Built-in tests for Disk and Printer status. Override for DOS critical error handler. Full contro over directories volume labels.
- Built-in 8086 assembler mnemonics for direct access to interrupts, controllers and ports.
- DOS PCL can be made RAM-resident and called up from anywhere with two keystrokes. Write your own resident applications or utilities with supreme ease.
- PCL needs to interpret a program only on the first pass, it generates extremely efficient machine code and re-uses it.
- In the OS/2 or the DOS developer's version you can save precompiled programs and eliminate the interpreters overhead altogether.
- With PCL's EXECUTE function your program can call the interpreter as a subroutine and execute source code immediately. This is useful for building intelligent self-modifying programs like Expert Systems, dynamic data dictionaries, spreadsheets, calculators etc.
- **PCL OS/2 Version 1 is available now.** It is upwards compatible with the DOS version, has unlimited program and data space and gives you an assured upgrade path to OS/2.

DOS PCL Single User	**£ 55**
OS/2 PCL Single User	**£ 80**
DOS PCL Developers Vers.	**£ 195**
OS/2 PCL Developers Vers.	**£ 220**
Upgrade from Single User	**£ 150**

PCL 3.0 comes on 5.25" or 3.5" diskettes with a 255 page comprehensive manual.

DOS PCL will run on any PC/XT/AT/PS2 and DOS 3.1 or above

OS/2 PCL runs in protected mode only and needs OS/2 Version 1.0 or above.

The developers versions include a royalty-free run time module and a source encryption utility for distributing your software to third parties.

Prices do not include VAT. Orders dispatched within 24 hours. All orders and enquiries to:

CALEND
P.O. BOX 94
TWICKENHAM TW2 6DD
Telephone: 01 894 7409 Fax : 01 755 0670

Environment

Despite the enormous power and facility provided by FoxPro, it runs happily in a 512 KB machine although the number of windows permitted will be restricted. On my Compaq 386/25 I started with free memory of 556 KB and, after loading FoxPro with expanded memory disabled, the MEMORY(0) command returned a value of 157 KB of available free memory, so the basic program load only required 400 KB. However, FoxPro will automatically make use of all LIM 3.0 or higher EMS it finds in the machine, (although you can specify how much EMS FoxPro is to use, or even totally disable it by setting EMS=OFF in the CONFIG.FP file). The more memory that is made available, the faster the program runs and the more windows you can define.

Most system defaults can be initialised through the use of the CONFIG.FP file. The interface uses a text based graphics screen, making extensive use of windows, default selection boxes and other features borrowed from the Apple Mac version of FoxBASE+. Although it can be used with a keyboard, a mouse is really mandatory to obtain the benefits offered by this attractive interface.

FoxPro is claimed to be fully FoxBASE+ and dBASE III/IV compatible, offering approximately 140 extended or new commands. Many of these are because FoxPro have provided the facility to add powerful parameters to such commands as 'Define Window' to allow added control by appending keywords such as CLOSE, FLOAT, GROW, SHADOW and ZOOM to the end of the normal command line. Some restrictions have been lifted. Command lines can now be 1024 bytes, and two dimensional arrays of up to 3600 elements are available. A surprising restriction in this release is that there are only 25 work areas available for database files, with a maximum of 25 index files open. (Clipper allows up to 255 work areas to be used for DBFs and indexes).

On initial load, a System Menu bar containing 7 options is displayed, of which six are actually available at this juncture. There is also a Command Window containing a cursor. Pressing F1 will produce on line help, although this is not necessarily context specific. Rather than the much maligned dot prompt, you enter commands into the command window, and any output is directed to the main screen behind the Command window. Naturally output can be redirected to other windows if required, but more of that later.

Support is provided for all normal video cards, using the commands SET DISPLAY TO EGA43/VGA50, which switches the display to 43 lines in EGA or 50 in VGA mode. This is very useful if you wish to run a standard 25 line application and also have trace and debugging windows visible in the additional lines. Other video standards are to be supported in the next release.

I was unable to use the demonstration programs provided with dBASE IV for comparative testing as the multiple Index files (.MDX) are not currently supported by FoxPro, but whilst speed comparisons between products such as these are always difficult to quantify, Fox claims of increases in the order of 8 times faster than dBASE IV and 16 times faster than dBASE III when using EMS memory did seem reasonable.

The Browse windows are amazingly powerful, allowing total control over every aspect such as the colour, size, and validation

The Product

From the familiar FoxBASE start-up screen, FoxPro allows you to enter commands directly into the Command window as if it were the dot prompt. All valid dot prompt commands can be used in this window.

Rather than being forced into a front end system such as the Assistant of dBASE III or the Command Center of dBASE IV to perform set-up and processing operations in a simple environment, all these facilities are available via a function call from a program or the System menu bar when working interactively. You can switch between windows in the same way as when running under Microsoft Windows, either by clicking on the window you want or by toggling through all visible windows with Ctrl-F1. Of course, the currently active window overlays any others. You cannot directly access the background screen, which is called the desktop, unless a program controlled data screen is there, but there is no reason to do so with the many facilities available from the windows, dialogue boxes and menus provided.

All system windows with the exception of the Dialogue panels can be defined to be totally flexible, provided that the appropriate permissions such as FLOAT, GROW and ZOOM have been given when the window was defined. They can then be sized, zoomed and moved at will, even by a user. Dialogue windows can be moved, but not sized or zoomed. Ctrl-F10 can also be used at any time to toggle windows to/from full screen display. Positioning the cursor on the dot at the bottom right of the window allows the window to be sized by dragging with the mouse whilst the whole window is moved by dragging it by the top panel. The flexibility of this approach allows easy control of many different operations at one time.

The seven System menu pads are System, File, Edit, Database, Record, Program and Window. Clicking on each pulls down a menu which can be made 'sticky' by use of a new SET STICKY ON/OFF command. When a Browse window is on the screen and active, an additional pad for Browse appears. To open a database, select the Window menu and click on the VIEW option. A dialogue window is displayed showing a table of work areas labelled A through J, with areas 11 through 25 being made available by scrolling the window down. On the right of the dialogue box is a table for relationships to be displayed and on the left, a set of eight useful generic options. These allow you to toggle the state of many of the system SET options, set up default drive and paths, change the currency symbol, punctuation, date formats and delimiters required, select or close databases and indexes in specific areas, and set up multiple browse windows. You can even control the mouse tracking speed.

With sufficient memory, up to 25 independent browsing windows can be on the screen simultaneously. Files may have relationships with other open files, and new relationships can be easily set up from within the View window. To do this, the Relations command is selected, and the parent file name chosen. When you select the child file, a powerful Conditions window is provided which allows you to build the most complex of relational expressions by selecting from scrolling windows of fields, systems variables, and any combination of the many FoxPro functions available. Multiple child relationships can be established.

ATTENTION dBASE USERS!

You could be programming in C tomorrow!

SYCERO

C

...is a powerful applications generator for Microsoft or Turbo C. Sycero generates fully structured and documented C code and lets you develop any type of application, from simple file maintenance programs to fully integrated commercial software. And you can program using dBASE, C or a mixture of the two!

Sycero C – breaking the language barrier!

System C

To - System C Ltd 60-61 High Street, Maidstone, Kent ME14 1SR Tel 0622 691616 Fax 0622 691241

Please send me further details about Sycero C, the C generator

Name Title Company

Address

Postcode

Tel Fax

dBASE is a trademark of Ashton Tate Microsoft C is a trademark of Microsoft Turbo C is a trademark of Borland

CIRCLE NO. 748

Browse Windows

The Browse. windows are amazingly powerful, allowing total control over every aspect including the colour, size, validation, and field usage. Once displayed, the size of the fields can still be adjusted with the mouse, and even the sequence of the fields displayed can be changed by simply dragging the field name to the new position required. Memo fields are displayed with the legend <Memo> if the field contains data, or <memo> if the field is empty. When you double click on these, the data in the current record Memo field is dynamically formatted and displayed in an editing window which is updated with data each time the memo field of any record is double clicked. This becomes the active window. Clicking again anywhere outside the Memo window leaves it on the screen but sets the Browse window as the current window. Calculated fields can also be defined as part of a Browse display.

Powerful field protection/validation is possible on all fields within a Browse window by using one of the many extensions to the FIELDS clause, such as /V. Using these, it is possible to perform Bounds checking, separate colour definition for each individual field, and conditional entry into a field dependent on its current contents. Fields can be set up to be read-only whilst others are editable, the displayed width can be controlled, and you can even call a UDF to perform complex validation by using the VALID (Function name) clause on each field. It is proposed in later releases to allow use of the PICTURE functions and masks

The CHANGE option from the BROWSE Menu allows the Browse window layouts to be toggled between the normal columns layout and a standard EDIT layout, which can use a FORMAT file if required. You can even split the browse window to display the data in both formats within the same window, giving linked or unlinked views of the currently selected record. This split can be varied in size by dragging the divider icon with the mouse.

You can have a BROWSE Window contained within your own windows as defined with the DEFINE WINDOW command, allowing your program to control what the user can do to it.

Memo fields

An important improvement is the access to those awkward Memo fields. Clipper first simplified access by giving us functions such as MEMOEDIT to access the data in a word processing environment. FoxPro has taken the concept to an elegant conclusion and now treats Memo fields identically to any other string variable, which can be up to 64 KB in size. This includes allowing operations such as sub-string searches, length checks, and conditional record selection based on the contents of memo fields. New commands such as `MODIFY MEMO` <Memo field name> have been implemented, which displays a full function editing window containing the data from the memo field in the currently selected record. If the window is re-sized whilst on the screen, the data is automatically reformatted to fit the window. There are several additional commands specifically for Memo field manipulation, such as `ATLINE`, `ATCLINE`, `MLINE` and `ATC` to access specific data in the memo, or to allow extraction of sub-strings of the data. The contents of a memo field can be written to a disk file with the command `COPY MEMO` <Field name> to <file name>, and also data read from disk to a memo field with the command `APPEND MEMO` <field name> `FROM` <file name>. You can even perform sub-string replacement of text in a memo field with a command such as `REPLACE SUBSTR(MEMFIELD,1,25) WITH SUBSTR(MEMFIELD,55,80)`.

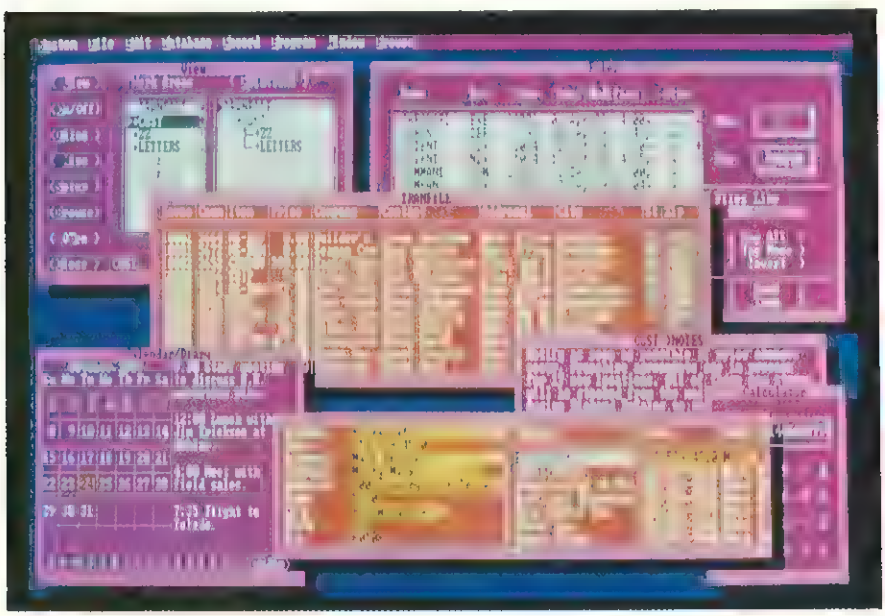
General features

Colour support is extensive. As well as the normal colour commands of dBASE, complete sets of colours for all objects may be defined and saved for use at a later date. If the SET BLINK OFF switch is used, then both foreground and background colours can use high intensity colours.

A full feature editor is provided, although by setting up `TEDIT` in the `CONFIG.FP` file, an editor of your own choice can be used in its place. Like every other area of FoxPro the built-in editor supports the mouse, allowing blocks to be cut, copied, pasted and moved. Global searches and replacements can be made, and the editor can be configured to control word wrap, tab sizes, indents, justification and even Control-Z sensitivity. In fact the cut, paste and copy operations are available throughout the system, allowing you to move or copy data between windows, to other fields or records in a browse window, or between the utilities such as the calculator and a database. The only complaint I have with this editor is the fact that there is no row/column indication, which makes the rather useful error files (.ERR) generated by the compiler unhelpful as you cannot find the line numbers quoted.

Keyboard macros are fully supported for virtually any combination of Ctrl, Alt, Shift and function keys, with the ability to save macro definitions to disk files for later use.

Programmer defined windows can be specified and then utilised freely to contain data displays, data entry screens or anything else. Windows have many attributes which can be individually defined, such as shadows on/off, initial window sizes, colours to be used, and permissions as to what operations a user of the application can perform in terms of sizing or moving the window once it is displayed. Multiple windows may be defined and then activated, deactivated or hidden as required, whilst a program can maintain full control of the



current output window. A window may even be hidden whilst still active, and program functions are provided to test for the status of any window. Activating a window will normally place it on top, but even this can be controlled if required. If data is sent to a window which is hidden, the data goes to that window just as you would expect, and if the window is later activated or 'SHOW'n, the data will be present in it exactly as you might have seen it appear had the window been visible. Window specifications can also be saved/retrieved from disk files for later reuse by programs.

When a window is activated, ALL output is directed to it, rather than the undetermined output destination provided by dBASE IV. If output is sent to a window that is not on top, you will see the output being performed correctly in any areas of the window that are still visible, whilst the topmost windows are completely undisturbed. The same is true if output is directed to the desktop itself.

Formatted output into an active window is positioned relative to the top left corner which is defined as row and column 0. This created various problems with existing code as the use of any co-ordinate with a `SAY. . GET` command that defines a position not visible in the window results in an error box being displayed with the message 'Position is off the screen'. By default, data is automatically wrapped by the window unless the system variable _WRAP is set to TRUE. It would perhaps be nice to see a true window facility with a command such as SET PAN to control the positioning of text. This could be useful as you could then let the user of an application zoom, size or pan the window if required to view any area of a fully windowed screen.

The full screen built-in report generator is powerful, allowing headers and footers to be defined with as many rows as you wish. Both fields and text can be duplicated or moved around simply by dragging with the mouse. Boxes can also be drawn around fields or text and any object automatically centred if required. Each item is treated as an independent object. Unfortunately there is a bug in the outputting of boxes to screen or printer, where the top line of all but the initial box is missing.

A powerful full-function DOS type Filer dialogue box is available from either the System menu bar or by use of the FILER command. All normal DOS operations such as copy, delete, rename, sort etc. are provided, and groups of files can be tagged/untagged. A tree of any disk drive can be generated, and paths/files selected from it.

There is also a simple calculator which accepts either keyboard or mouse input; a calendar/diary which can retain notes on appointments and clearly shows from the monthly date display what dates have appointments in them; an ASCII look-up chart and finally a useful screen capture facility that grabs any marked area of the FoxPro screen to the Clipboard, which can then be pasted anywhere in the system, including into database fields or memo fields. Finally there is a shuffle puzzle for when you get bored with working.

FoxPro runs happily in a 512 KB machine although the number of windows permitted will be restricted

I did encounter a few surprising omissions, such as the lack of a descending index order facility and no SET CURSOR ON/OFF toggle, although many of these are supported through the use of the original FoxBASE+ SYS() functions. This might make conversion of existing systems from dBASE or Clipper more complex than necessary.

The READ statement can now be truly nested, which is extremely useful when you wish to jump from, say, an order entry screen to a customer details input screen, enter the details of the customer and then return to the order entry system without having cleared any pending GETS from the order entry system.

With release 1 of the product, Fox are supplying copies of the original FoxBASE+ utilities FoxView (a screens design tool), FoxCode (an Applications Generator), FoxDoc (a documentation tool) and Foxgraph, which unfortunately is only provided as a crippled demo version and you have to buy the full version separately to benefit from the 3D graphics facilities it is able to provide.

While this release does not provide a true compiler, I understand that this is to be available with the next release, which is apparently to be provided free of charge to all purchasers of version 1. At the moment, FoxPro still uses the familiar tokenising compiler from FoxBASE+, which generates the interim code run by the system. The true compiler is promised in the first half of this year and will generate stand alone executable files in the same way as Nantucket's Clipper. It is said that this will fully support external functions written in 'C' and other languages, but unlike Clipper it is claimed that all stack and memory management for this interface will use the C standard rather than the more complex extensions and hooks to the Library provided by Nantucket to pass and receive parameters and generally manage the non standard stack used.

Conclusion

Overall, this product is the most powerful yet in the competitive dBASE language market place. Its richness of facility and power, together with speed and flexibility will, I am sure, make it a major contender in this most competitive area of the software market. Fox Software have confirmed that they are working on a SQL interpreter in addition to the true compiler, and both are scheduled for early release.

Fox have shown great technical expertise as well as innovation in producing this original approach to the dBASE Environment. I suspect that this product will make the Ashton-Tate / Fox battle even more interesting. If the true-executable compiler is as good as Clipper perhaps even Nantucket will need to look carefully at their soon to be released version 5.0 of Clipper if it is to retain its hold on the professional dBASE compiler marketplace. Serious developers would be well advised to take the time to evaluate this product, particularly now that Ashton-Tate have publicly accepted that their own dBASE IV compiler is a long time away. .EXE

Ian Turner has been closely involved with the dBASE language for many years, having originally been the UK Technical Director of Ashton-Tate in its formative years here in the UK and Europe.
After leaving Ashton-Tate he was involved in the introduction of the Clipper compiler to the UK before starting his own Consultancy, working mainly with dBASE, Clipper, FoxBASE and the C Language, as well as desktop publishing products, and he has done considerable applications work using these products in corporate organisations in both the UK, Europe and the USA. He can be contacted on 0908 670996.

Hardware Focus

It's time to come clean. Although 98% of our readers use PCs, most of you felt that reviewing them was not what .EXE should be doing. To make amends (hopefully), this issue sees the start of Hardware Focus – stories about hardware, but with a definite software bias.

IBM launches 486-based PC

IBM has launched a 486-based version of its PS/2 Model 70. The machine, which at present is available only in the US, is the second IBM system to be based around the 486. The first was the 486 Power Platform, a 486-on-a-card which replaced the 386 processor in a standard Model 70. The new PS/2 appears to be a Model 70 with the Power Platform already installed. It remains to be seen whether a 386 box with an upgraded processor can perform as well as a box built from scratch around the 486.

US price for the Model 70/486 is $12,390, with 60 MB hard disk, which is no different to the price of a Model 70/386 with the Power Platform installed. However, buying a Model 70 with the Power Platform already installed, means that your dealer misses out on a free 386 chip and a couple of BIOS ROMs, which he normally gets to keep when installing the add-in board.

CIRCLE NO. 780

486 In Production

Intel confirmed shortly after Christmas that the 486 is now in production, though there are no details of how many chips are available or where they are going. The final revision, known as mask B6, is said to cure all known bugs, and should remove the problems faced by some software packages which used undocumented areas of the 386 for their own purposes.

The chip is available in a 25 MHz version at present, though Intel stated at the launch that 33 and 45 MHz versions would appear within a year or so. However, recent reports have highlighted major reliability problems with 33 MHz 386 systems, and it remains to be seen whether the 486 will be able to overcome these.

CIRCLE NO. 781

New Microsoft Mouse

Microsoft has launched a new mouse, with a new release of the driver software. As before, the 400-Series Mouse comes in bus and serial varieties, and an OS/2 version of the driver software is now provided as standard. The resolution has been doubled to 400 points per inch, and the serial-to-PS/2 interface module has been eliminated.

A new pop-up control panel provided by the driver software allows more control over mouse acceleration. There are now 16 acceleration levels, allowing you to specify different cursor movement speeds on screen to correspond to varying speeds at which the mouse is pushed across the desk. The driver software reads its settings from a MOUSEPRO.FIL text file, and different settings can be made for different application programs. This allows users to set up special configurations that will zoom quickly around a large spreadsheet, without the fear of running off the end of a small WP document.

From a software point of view, the new mouse driver is functionally identical to existing versions. Microsoft UK were not aware of any new functions or commands that had been added to the standard Int 33h interface.

Microsoft constructed a special version of the mouse driver for in-house use, during the design of the 400-Series mouse. The driver kept a permanent record of every mouse movement, and could provide statistics on how far the mouse had travelled, in what direction, and the total number of times that each of the mouse buttons had been pressed. If you want to amaze your friends at a dinner party, forget the one about 'when you sneeze, it comes out your nose at over 100 miles an hour', and tell them that a 'heavy user' (who works with Windows packages most of the time) rolls his mouse around 60 feet per day, while the occasional user only manages around 66 inches.

The new mouse costs £125, bundled with Paintbrush, and Mouse Menus, or £150 if bundled with Windows/286 and Paintbrush For Windows. If you want to convince Microsoft that a mouse driver with built-in tachograph would be a wonderful product to sell, call them on 0734 391123.

CIRCLE NO. 784

Higher Res Monitor

If you really want to show off your latest creations, get hold of the latest monitor from Megascan. This 19-inch screen provides a monochrome resolution of 4096 by 3300, which is almost as good as the 300x300 dots per inch on a standard laser printer. Details from CGI, the UK distributors, on 0672 20777.

CIRCLE NO. 782

860 Product Launched

Intel launched the i860, its most powerful processor, in the middle of last year. IBM has now launched an i860-based system which is usable from a PC environment.

The IBM PS/2 Wizard Adaptor is an MCA bus master card (a card with a CPU on it). It's designed to slot into a PS/2, and contains an i860, 2 MB of RAM and a couple of custom IBM chips. The 860, a RISC processor that's not compatible with the 80x86 family, runs at 33 MHz, and IBM are claiming benchmark figures of 64 Mflops and 27 Vax MIPS.

To drive the card, you need the imaginatively-named 'The Intel i860 Microprocessor OS/2 Software Development Tools' pack, which contains a library of C routines that allow whole programs , or parts thereof, to be moved across to the Wizard card, executed and, if necessary, results returned.

Cost of the card is £5568, and the software toolkit is another £1168. If 2 MB of RAM is not enough, you can expand it to 8 MB with the Memory Expansion Option, for £2915. Total price, £9651.

CIRCLE NO. 783

If you would like more information on the hardware news stories above, please turn to the Free Reader Information Service Card in this magazine and circle the relevant circle numbers.

Manufacturers or distributors with products that they feel would be suitable for inclusion on this page, should send full details to The Editor at the address on page 2, or fax them to 01 994 1533.

BEST BUY
"So simple to use you wonder why other packages make such a meal of resource allocation."

PC MAGAZINE

"InstaPlan offers the features of a £2,000 package with none of the trade offs of its low cost competitors."

PERSONAL COMPUTER WORLD EDITOR'S CHOICE

"We would recommend most warmly... InstaPlan as the best featured project management package."
Practical Computing
(compared with Project Manager Workbench, Super Project Expert and Harvard Project Manager).

"InstaPlan is excellent, much better value than PMW, but keep improving it" Mr D Symington, CS Consultants

NOW InstaPlan 5000

InstaPlan 5000 is the latest addition to the InstaPlan family of project management software, which already has 20,000 users, from large multinationals to one man businesses, and is fully compatible with the rest of the InstaPlan family.

Easier Departmental Management

InstaPlan 5000 can gather individual project plans and consolidate them for departmental review, all with a single command on a stand alone machine or a PC network.

By bringing together the department's schedules and resource requirements in one model, you can recognise and resolve staffing conflicts and project priorities long before they become crises.

Most project management systems require all projects to be updated to the same data before merging. This is often impossible. InstaPlan 5000 allows projects to be combined freely.

Planning and Control for Larger Projects

InstaPlan 5000 will run on a standard 640K PC, Just like the rest of the InstaPlan family. But InstaPlan 5000 also supports EMS (Expanded Memory). This means that you can handle in excess of 10,000 Activities. Within these limits you are only constrained by the amount of EMS memory you have available. This is many times the capacity of some of InstaPlan's more expensive competitors.

Better Resource Management

You can now level the load on a group of resources simultaneously, using he Adjust Loading command. The Locate Overload facility will scan all future periods for a resource, identifying any overloads.

Links with Lotus 123 and dBASE

With InstaPlan 5000 you can edit plans for export in Lotus 123 and dBASE form. This makes it easier to produce summary level graphs and pie charts. We believe that InstaPlan 5000 provides a solution both for professional project managers and for busy line managers who need project management techniques to enhance their own and their department's performance.

See what InstaPlan users say:-

Easy to learn and Use

...it's design is just right. I set up a workload plan without even using the manual or any previous planning experience.
David Allen — *Computer Services Manager*
Vallectric Ltd — *Electrical Contractors*

With Quality Documentation

...a very good package, with excellent documentation.
Mr J Clarke — *Finance Director*
Ceraf Ltd — *Building Contractors*

...And Support too

...Good telephone support
Dr Robin Wilson
Harrogate Borouch Council

Providing the Right Reports

...provides me with succinct and easy to understand documentation for Board level presentations.
Dr R Galley — *Director of Research*
Tootal Thread Ltd

For Line Managers

Even though our busy line managers are only occasional users, InstaPlan gives them the tools to plan quickly and effectively.
Brian Alldridge — *MIS Manager*
Prestwich Holdings plc

and for Project Managers

I am surprised to find myself learning from this product – an excellent investment.
Mr P M Cleary — *Project Manager*
Maritime Dynamics Ltd

compared with the competition

Having suffered with Project Manager Workbench and SuperProject Expert, InstaPlan has provided us with the tools to produce all our reports and regular project updates easily. Management are impressed by the ease with which they can read and understand the reports.
Mrs Petra Meyer — *Product Manager*
Travicom Ltd

In summary

InstaPlan contains all the features and facilities required for project tasking, costing and tracking. An excellent, good value package, it forms the basis of successful project management within a business environment.
Andy Coster
QA Manager, Logsys

Our Introductory Offer

InstaPlan 5000 is £499
However, order before 31st December 1989, and you benefit from InstaPlan 5000 for only £449, a saving of £50

Money Back Guarantee

We are confident InstaPlank 5000 is the easiest project management system to learn and use, offering a significant price/performance advantage over all other systems. Buy InstaPlan 5000 now, if you return it within 30 days for whatever reason, we guarantee to refund your money by return.

Configuration. IBM XT/AT PS/2 or compatible running under PC/MS DOS 2.1 or later with at least 500K of DOS memory and 3M of hard disk avialable to InstaPlan. Plan storage requires further DOS memory or access to EMS memory. EMS memory (LIM3.2+) supported to a maximum of 8Meg. A 5000 activity plan will require at least 2 Meg of expanded memory. InstaPlan 5000 will therefore run across networks using network shells like Novell where there is sufficient DOS memory available to InstaPlan. Printers: IBM ProPrinter & Graphics Printer, Epson FX/LX/LQ, Toshiba 351, HP LaserJet PLUS/Series II Deskjet. Input via keyboard, ASCII, ThinkTank or READY! Output to ASCII, Lotus or dBase. Charts are stored as bit mapped graphics and can be printed but not viewed on screen. ALL MARKS ACKNOWLEDGED. DEALER ENQUIRIES WELCOME. QUANTITY DISCOUNTS AND SITE LICENCES AVIALABLE. TRAINING AND CONSULTANCY AVIALABLE IF REQUIRED

DEEPAK SAREEN ASSOCIATES

Freepost 28 Preston Rd, Wembley HA9 8BR
Phone 01 908 2423 Fax 01 904 6668

InstaPlan 5000 £449 plus £6 p&p and VAT

- ☐ Please send me InstaPlan 5000.
- ☐ I enclose a cheque for £523.25 or
- ☐ Please bill my Visa/Access Card
- ☐ send me more details of InstaPlan 5000

Company.....

Name.....

Signature.....

Phone no.....

Please enclose a compliments slip showing your required delivery address and card details if appropriate.

.EXE 2.90.

CIRCLE NO. 749

The DOS Keyboard Buffer

Expanding the DOS type-ahead keyboard buffer beyond 16 bytes is best done with a device driver rather than a normal TSR. Paul Jenkins explains why and how.

Ever since I started using an IBM PC, it has struck me that the standard type-ahead keyboard buffer is just that bit too small. All of us have experienced that annoying beep at some time, when trying to enter a long DOS command while the PC's busy working.

On paper, replacing the keyboard buffer with a longer one looks reasonably easy. The buffer normally resides in low memory (in segment 40h), and two 16-bit pointers are used by the BIOS to keep track of the head and tail positions of the buffer. To make a longer buffer, you simply reserve some space in memory, and then change the two pointers so that they now point at your extended buffer, rather than the original one.

Unfortunately, there's a problem. The pointers are, as I have said, only 16-bits wide. You can't move the buffer out of segment 40h because this segment value is hard-coded into most BIOS routines. When DOS loads, the DOS itself goes into RAM first. On top of this go the device drivers that are loaded from CONFIG.SYS, and then come the TSRs from AUTOEXEC.BAT.

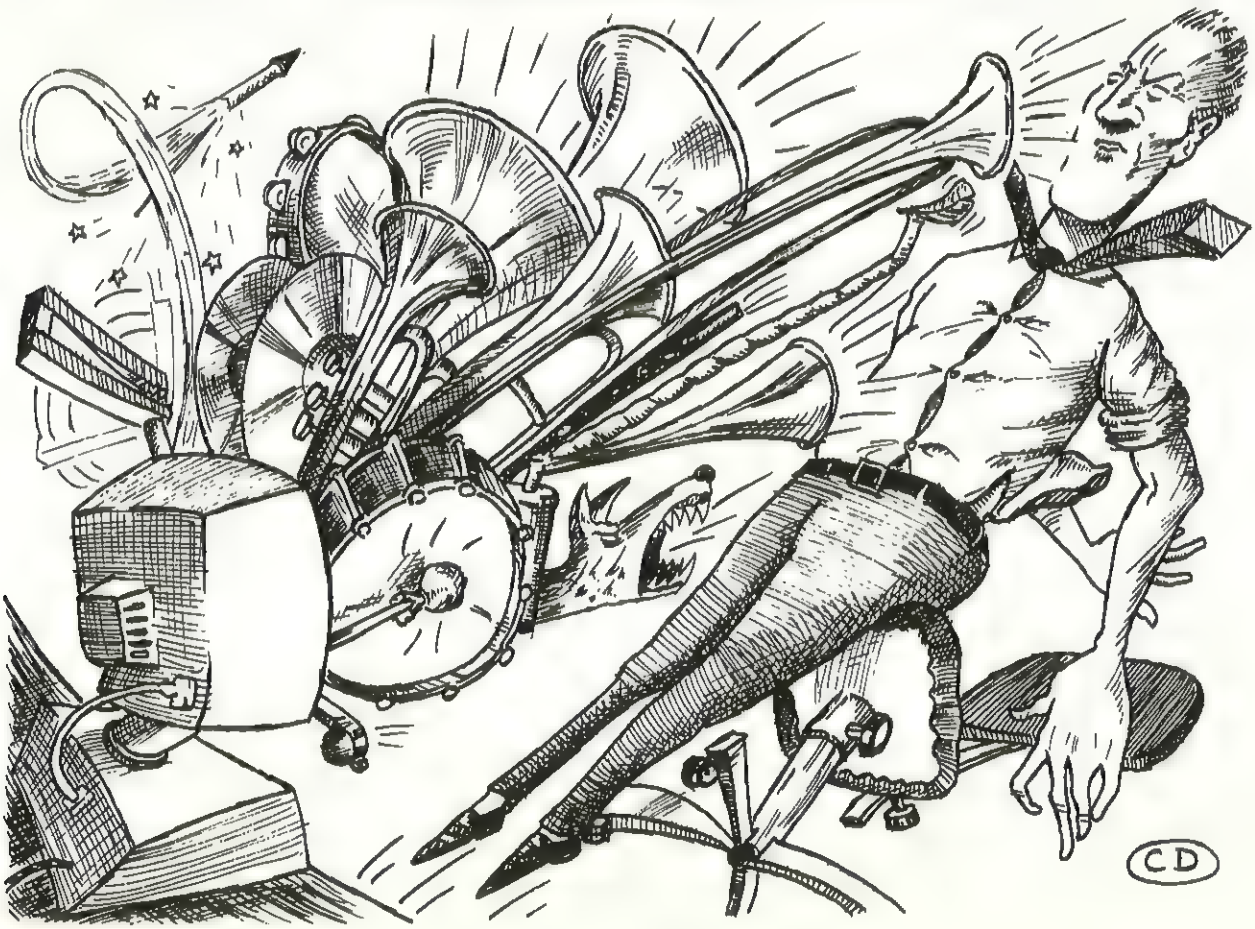
By the time DOS has loaded, and set up a standard number of FILES and BUFFERS, and your mouse driver has been installed, chances are that there is no free memory in segment 40h, and so it's impossible to expand the keyboard buffer in a way that will work with almost all software. DOS 4, by the way, takes up so much RAM that, even without any FILES or BUFFERS or device drivers or TSRs, there is no low memory left.

To get round this problem, I implemented my keyboard buffer expander as a device driver, rather than as a TSR. Device drivers get loaded comparatively early in the boot process, and there is more chance of success if you adopt this route. Also, the program below is sufficiently short to serve as a useful example of a working MS-DOS device driver.

How it Works

DOS invokes a device driver by making successive calls to the driver's Strategy and Interrupt routines. The Strategy routine simply takes the pointer to a Request Header block passed to it by DOS, and saves it within the device driver. The Interrupt routine uses this pointer to access the Request Header and determine the function required.

Our initialisation code first checks that the new keyboard buffer can be located entire-



CALL FOR FREE 80 PAGE CATALOGUE

ASSEMBLERS

386 ASM Link & Locate	£435.00
MASM v.5.1	£89.00
OPTASM	£79.00
Turbo Assembler/Debugger	£59.00

BASIC

Microsoft Professional BASIC	£320.00
BASIC Interpreter	£215.00
Quick Basic	£60.00
True Basic	£65.00
Turbo Basic	£65.00

C

Lattice C Compiler v.6.0	£160.00
Microsoft C Compiler v.5.1	£250.00
Watcom C Optimizing	£245.00
C-terp	£189.00
C86 Plus	£310.00
Eco C88 with dev. library	£55.00
Express C	£60.00
High C-286	£375.00
High C-386	£560.00
Instant C	£310.00
Instant C/16M	£495.00
NDP C	£375.00
Quick C	£65.00
Turbo C v.2.0	£89.00
Turbo C Professional	£159.00
UserSoft/C	£150.00
Zortech C Compiler	£45.00

C++

Advantage C++	£310.00
Guidelines C++ with book	£185.00
Intek C++ 80386	£310.00
Zortech C++ Compiler v.2.0	£120.00
Zortech C++ Dev. Edition	£290.00

COBOL

acucobol MS-DOS	£995.00
acucobol Net-Bios	£995.00
acucobol XENIX-286	£995.00
acucobol XENIX-386	£995.00
acucobol OTHERS	CALL
Microsoft COBOL	£565.00
Real Menu	£210.00
RealCICS	£750.00
Realia COBOL	£750.00
RM/Cobol	£575.00
RM/Cobol-85	£750.00

dBASE

Clipper Version 5.0 U.K. *New*	£325.00
DB Fast/PLUS	£155.00
DB Fast/WINDOWS	£155.00
dBASE III Plus	£395.00
dBASE IV	£395.00
dBMAN V	£120.00
dBXL Diamond	£155.00
dBXL/Lan	£375.00
Foxbase+ v.2.10	£245.00
Foxbase+/386	£375.00
Foxbase+/LAN	£375.00
Foxbase+ Runtime	£435.00
Foxbase+ SCO XENIX	£495.00
Quicksilver Diamond	£375.00

DATABASE MANAGEMENT

Advanced Revelation	£595.00
AskSam	£185.00
Clarion Personal Developer	£99.00
Clarion Professional	£435.00
Omnis Quartz	£480.00
Opus One	£240.00
Oracle for 1-2-3	£120.00
Paradox 2.0	£439.00
PFS Professional File	£149.00
R:BASE Ext. Report Writer	£179.00
R:BASE for OS/2	£545.00
R:BASE for DOS	£439.00
Reflex	£89.00

FORTRAN

Microsoft Fortran v.5.0	£285.00
Fortran-F77L	£299.00
NDP FORTRAN-386	£360.00
ProFortran	£235.00
RM/Fortran	£375.00
SVS Fortran	£435.00

OTHER LANGUAGES

ACTOR	£310.00
ACTOR Language Extension	£60.00
CCSM Mumps Single User	£65.00
PC Logo	£90.00
RPG II Complete	£995.00
Smalltalk/V 286	£125.00
Snobol 4+	£60.00
Trilogy	£60.00

PASCAL

Microsoft Pascal	£189.00
Oregon Pascal-2	£149.00
ProPascal	£235.00
Metaware Professional Pascal	£560.00
Metaware Prof. Pascal 286	£375.00
Turbo Pascal v.5.5 with objects	£89.00
Turbo Pascal Professional	£159.00

PROLOG

Arity Combination Pack	£685.00
Expert System Dev. pack	£185.00
Intelligence Compiler	£299.00
MicroPROLOG Professional	£425.00
MProlog	£299.00
Professional Prolog II Plus	£899.00
Professional Prolog	£545.00
Prolog Advanced Toolkit	£90.00
Prolog Compiler	£120.00
Prolog Compiler/Interpreter	£399.00
Prolog-86 Plus	£150.00
SQL Development Pack	£185.00
Turbo Prolog	£75.00

AI AND OBJECT ORIENTED

C-Talk	£95.00
C+ Objects Vol 1 with Source	£275.00
CxPERT	£245.00

COMMUNICATIONS

AdComm	£185.00
C ASYNCH MANAGER	£110.00
Communications Library	£145.00
Comm Pak with Breakout	£195.00
Multi Comm	£89.00
Net Lib	£125.00
NET-TOOLS	£95.00
SilverComm	£120.00
Comm Plus	£60.00
Turbo ASYNCH PLUS	£80.00

FILE MANAGEMENT

asmTREE	£245.00
B-Tree with Source	£110.00
B+Tree for C	£49.00
AXOS	£245.00
Btrieve v.5.01 Novell	£155.00
Btrieve OS/2	£370.00
Btrieve Multitasking	£370.00
Btrieve/N v.5.01	£370.00
Btrieve XENIX	£370.00
C Index+	£245.00
c-tree	£245.00
Code Base 4	£185.00
db_FILE/RETRIEVE	£375.00
d-tree	£310.00
dBC III	£160.00
dBC III Plus	£320.00
RealDL/I	£599.00
WKS Library	£125.00

GENERAL

ASMLIB	£89.00
C Tools Plus	£79.00
C Utility	£120.00
C*Lib C Function Library	£120.00
Clipper toolbox	£239.00
DB/RA3	£125.00
dBase Program Utility	£55.00
dBase Program Utility Vol 2	£55.00
dBase Tools For C	£55.00
Finally	£60.00
Functions with source	£125.00
Integrated Dev. Libraries	£90.00
Pascal Toolkit	£90.00
Pascal Tools 1	£75.00
QuickPak Professional	£90.00
Quick Tools Library	£79.00
Spindrift Library	£90.00
Super Functions	£160.00
Tom Rettig Help Combo	£75.00
Tom Rettig Library	£60.00
Turbo C TOOLS	£79.00
Turbo Plus	£60.00
Turbo PowerTools Plus	£79.00
Turbo Functions	£69.00

GRAPHICS

dBase Tools for C Graphic	£55.00
dGE v.3.0 *NEW*	£125.00
EGA Toolkit	£149.00
Essential Graphics	£180.00
Flipper	£120.00
FontWINDOW/Plus	£170.00
GEOGRAPH Professional	£239.00
Grafmatic CGA & EGA	£135.00
GraphiC	£245.00
GSS Graphics Dev. Toolkit	£389.00
GSS Kernel	£299.00
GSS Metafile Interpreter	£179.00
HALO-88	£245.00
HOOPS	£360.00
HOOPS 32	£349.00
MetaWINDOW/Plus	£169.00
scLASERplus	£299.00
SilverPaint	£60.00
SilverPak	£179.00
Slate	£215.00
Universal Graphics Library	£130.00

OP. SYSTEM ENHANCEMENTS

C Sharp	£300.00
DESQview API C Lib/Ref	£125.00
Multi C	£89.00
Multi Dos Plus	£60.00
OS/2 Programmer's Toolkit	£215.00
OS/286	£299.00
OS/386	£299.00
Resident C	£120.00
Sapiens V8	£180.00
Security Lib with Source	£150.00
Synergy Toolkit	£239.00
Time Slicer	£179.00
VMFM	£115.00
Wendin OS Toolbox	£60.00
Wendin PC UNIX	£60.00
Wendin PC VMS	£60.00
Windows Development Kit	£295.00

QUERY

CBTREE/SQL	£239.00
CQL Query System	£239.00
dQUERY/Lib	£179.00
db_RETRIEVE	£239.00
XQL	£480.00
Xtrieve PLUS	£360.00
Netware SQL	£385.00

REPORT

CB/IQ Report	£139.00
Coreports	£179.00
Focus Report Writer	£210.00
r-tree	£179.00
R&R Relational Reportwriter	£95.00
R&R Clipper Module	£49.00

TEXT SCREENS

Animator COBOL	£120.00
Application Manager	£450.00
C-Worthy v.1.1	£160.00
C-Worthy + Forms	£240.00
C-Worthy + Forms + Source	£395.00
C-Worthy All C compilers	£480.00
C-Worthy Forms only	£80.00
C-Worthy Source code only	£155.00
C-Worthy OS/2	£240.00
C-Worthy + Forms for OS/2	£320.00
C-Worthy + Forms + Source	£480.00
C-Worthy OS/2 & MS-DOS	£315.00
C-Worthy + source	£795.00
Curses	£150.00
Curses Window Dev. Kit	£75.00
Data & Windows	£149.00
DataWindow	£179.00
dWindows for db3	£60.00
Face-it	£60.00
Flash-up w/Toolbox	£85.00
Forms II	£179.00
HI-SCREEN XL	£89.00
Multi Windows	£179.00
On-Line Help	£90.00
PANEL PLUS	£299.00
PC Forms	£90.00
Pop Screen	£55.00
POWER SCREEN	£79.00
Pro Screen	£60.00
SCREENIO	£245.00
Screen Sculptor	£75.00
ScreenStar	£120.00
Screenplay	£110.00
SoftCode	£120.00
SoftCode Template	£30.00
Softscreen/Help Source	£150.00
Sourcewriter	£605.00
UI Programmer	£179.00
V Screen	£39.00
VCScreen	£90.00
Vitamin C	£135.00
VLib with Source	£90.00
Vermont Views	£245.00
Z View	£149.00

DEBUGGERS & DISASSEMBLERS

386 Debug	£125.00
Breakout-II	£79.00
D'Scope 2	£159.00
dBug for dBase	£125.00
DIS*DOC	£79.00
Periscope II	£110.00
Periscope II-X	£90.00
Periscope III	£875.00
Periscope I Rev 3	£435.00
Periscope IV (16 MHz)	£1245.00
Periscope IV (20 MHz)	£1435.00
Periscope IV (25 MHz)	£1625.00
Plus Board with 512K	£310.00
Plus Board with 0K	£220.00
Serial Test	£310.00
Soft Probe II/TX	£245.00
Soft-X-plore	£79.00
Sourcer + BIOS Pre-Processor	£89.00

DOCUMENTORS

ASM Flow	£60.00
Clear+ for dBASE	£125.00
Clear+ for C	£125.00
dAction	£55.00
dAnalyst	£139.00
Design/2.0	£215.00
dFLOW	£89.00
Documenter	£179.00
Flow Charting II Plus	£139.00
FoxToolBox	£179.00
PolyDoc	£169.00
PolyXREF (All languages)	£60.00

EDITORS

BRIEF v.3.0 *NEW*	£149.00
BRIEF v.3.0 *UPGRADE*	£60.00
BRIEF OS/2 (U.K. Version)	£149.00
dBRIEF v.2.1 (U.K. Version)	£79.00
BRIEF with dBRIEF	£219.00
C-EDIT	£239.00
EMACS	£199.00
Epsilon	£110.00
ME Editor with C Source	£115.00
Micro ISPF	£110.00
Norton Editor	£49.00
PC/EDIT Plus	£179.00
Personal Rexx	£79.00
PI Editor	£119.00
SPF/PC	£149.00
Turbo Basic Editor Toolbox	£59.00
Vedit+	£115.00

HELP GUIDES AND TRAINING

dBASE on-line	£60.00
Norton Guides for ASM	£75.00
Norton Guides for BASIC	£75.00
Norton Guides for C	£60.00
Norton Guides for OS/2	£89.00
Norton Guides for Pascal	£75.00
Programmer's Library	£239.00

LINKERS & LIBRARIANS

ALINK *NEW*	£130.00
Archive Link	£239.00
Link & Locate ++	£245.00
LinkLok	£245.00
PLINK86 Plus *Now Polytron*	£495.00
PolyLibrarian I	£85.00
PolyLibrarian II	£125.00
.RTLink	£180.00

MAKE UTILITIES

PolyMake v.3.2 *NEW*	£125.00
MKS Make	£85.00

PROFILERS & OPTIMISERS

Codesifter	£75.00
DOSTRACE	£59.00
PC Metric	£120.00

PROTOTYPERS & GENERATORS

Automated Programmer	£605.00
Cogen	£275.00
COMPEDITOR	£110.00
Dan Bricklin Demo II	£125.00
Data Boss for C	£245.00
Data Boss for Pascal	£245.00
Genifer	£239.00
Instant Replay III	£89.00
Matrix Layout v.2.0	£129.00
Sycero db (U.K.)	£595.00
Sycero db Net (U.K.)	£745.00

SYNTAX ANALYSERS & LINTS

PC Lint v.3.0	£89.00
Pre-C	£150.00

TEXT & UNIX TOOLS

C Programmers Combination	£79.00
C ToolSet	£59.00
MKS LEX & YACC	£149.00
MKS Toolkit v.3.1 *NEW*	£149.00
MKS Toolkit v.3.1 *UPGRADE*	£50.00
PC Fortran Tools	£99.00
PC Yacc	£249.00
PolyAWK	£60.00
PolyAWK OS/2	£115.00
QPARSER Plus	£180.00
Text Management Utilities	£75.00
Turbo Professional	£75.00

TRANSLATORS

BAS_C Commercial	£229.00
Bastoc	£480.00
dBx dBASE to C v.4.0	£345.00
dBx Library Source	£250.00
dBx UNIX/XENIX Source	£470.00
FOR_C v.2.0 MS-DOS *NEW*	£565.00
FOR_C v.2.0 XENIX *NEW*	£630.00
FOR_C++ XENIX/386 *NEW*	£1195.00

VERSION CONTROL

PVCS Personal v.2.1c	£85.00
PVCS Professional v.3.1	£395.00
PVCS Network (5 User)	£1395.00
PVCS/PolyMake OS/2	£CALL
PVCS/PolyMake VAX/VMS	£CALL

BACKUP - RECOVERY - REPAIR

dSalvage	£59.00
Fastback	£105.00
Fastback Plus	£115.00
Mace Utilities	£59.00
Norton Utilities Advanced	£89.00
PC Tools Deluxe	£49.00
XTree Net	£235.00

CACHE

PolyBoost II	£49.00

COMMS. & TERM. EMULATORS

Blast Communications	£149.00
Carbon Copy Express	£199.00
Carbon Copy Plus	£115.00
Crosstalk Mark 4	£145.00
Crosstalk XVI	£115.00
Mirror II	£45.00
PC Anywhere III	£85.00
Sidetalk II	£70.00
SmarTerm	£75.00
SmarTerm 125	£175.00
SmarTerm 220	£115.00
SmarTerm 2392	£135.00
SmarTerm 240	£199.00
SmarTerm 400	£99.00
SmarTerm 4014	£135.00
Z Stem PC-VT100	£59.00
Z Stem PC-VT220	£89.00
Z Stem PC-VT240	£175.00
Z Stem PC-VT4014	£59.00

DISK MANAGERS &OPTIMISERS

HTest/HFormat	£59.00
Mace 5	£65.00
Mace Gold	£95.00
Norton Utilities Advanced	£95.00
SpinRite II	£59.00

OP. SYSTEM ENHANCEMENTS

386MAX	£49.00
386MAX Professional	£80.00
DESQview	£79.00
Norton Commander	£55.00
Omni View	£49.00
PolyShell	£60.00
QDOS II	£45.00
Soft Bytes 386 Max	£45.00
Weiner Shell	£175.00
Power Bundle	£99.00
Windows/286	£59.00
Windows/386	£115.00
X-Tree Pro	£75.00

FAX OR TELEPHONE CREDIT CARD ORDERS VISA OR ACCESS

PRICING
All prices exclude V.A.T., but include first class recorded delivery.

EDUCATIONAL DISCOUNTS
Generous Educational discounts available on certain products.

CATALOGUE & FACT SHEETS
Call for our new Winter '90 catalogue, over 80 pages of Software Development Tools.

PCW 02.90

THE SOFTWARE CONSTRUCTION CO. LTD.

THE RED HOUSE, 84 HIGH ST., BUNTINGFORD, HERTS. SG9 9DJ (0763) 73455 FAX. (0763) 73460

CIRCLE NO. 750

ly in segment 40h. Assuming that it can, the driver changes the four pointers associated with this buffer. The relevant addresses, as offsets in segment 40h, are:

- 1Ah – read pointer.
- 1C – write pointer.
- 1E – 16-word circular keyboard buffer.
- 80h – pointer to start of keyboard buffer.
- 82h – pointer to end of keyboard buffer.

This new expanded keyboard buffer should be compatible with all programs which use DOS and BIOS function calls or even manipulate the variables listed above. Only those applications which blindly assume the use of the standard keyboard buffer will cause problems. It is impossible in a program of this size to cater for such 'ill-behaved' applications. EXE

Paul Jenkins is a freelance computer programmer, and can be contacted via the editorial office, or on Cix as PJenkins.

```
; This device driver implements an 80-character type-ahead keyboard buffer in low memory
; (in the same 64K segment as the default DOS buffer) on a PC.  This device driver should
; be the first entry in CONFIG.SYS to ensure successful installation.

code    segment
        org 0     ; Required for a device driver
        assume cs:code, ds:code

; These are the device driver variables
Start:  dd   -1                         ; (null) pointer to next device driver
        dw   8000h                      ; character device, non-IOCTL
        dw   Strategy                   ; pointer to the first procedure
        dw   Interrupt                  ; pointer to the second procedure
        db   'PMJ-1989'                 ; 8-byte string of unique device name

; These are the program variables
        even                            ; Ensure buffer etc. is on word boundary
RHseg   dw   ?                          ; Request Header segment
RHoffset dw  ?                          ; Request Header offset
BuffStart dw 80 dup (?)                 ; New keyboard buffer

; The STRATEGY procedure stores location of the Request Header passed to it by DOS in ES:BX.
Strategy: mov cs:RHoffset, bx           ; Save Request Header offset
          mov cs:RHseg, es              ; Save Request Header segment
          retf                          ; Far return

; The INTERRUPT procedure processes the command in the Request Header.  The
; following fields in the Request Header are relevant to this device driver:

;    Byte      Content
;    0h        Length of Request Header in bytes
;    2h        Command ( 0 = Initialise )
;    03h-04h   Status ( bit 8 = done )
;    0Eh-11h   Free memory address

Interrupt: push ds                ; Save all registers
           push es
           push ax
           push bx
           push cx
           push dx
           push si
           push di
           push cs                      ; Make local data addressable
           pop  ds
           mov  bx, RHoffset            ; Point ES:BX to Request Header
           mov  es, RHseg
           cmp  byte ptr es:[bx+2], 0   ; Command = Initialise ???
           je   Init                    ; If so perform initialisation

Exit:      or word ptr es:[bx+3], 100h  ; Return Status=Done regardless
           pop  di                      ; Restore all registers
           pop  si
           pop  dx
           pop  cx
           pop  bx
           pop  ax
           pop  es
           pop  ds
           retf                         ; Return to DOS
; This is the initialisation code. Free memory starts here when installed
Init:      mov  word ptr es:[bx+14], 0  ; Indicate failure to install
           mov  word ptr es:[bx+16], cs ; until proved otherwise
           mov  ax, 40h                 ; Set up DS with segment of default buffer
           mov  ds, ax
           mov  ax, cs                  ; Convert CS segment address of this
           sub  ax, 40h                 ; device driver into the form 0040:xxxx
           cmp  ax, 0fe0h
           ja   Exit                    ; Quit if start address outside segment
           mov  cl, 4
           shl  ax, cl
           add  ax, OFFSET BuffStart
           jc   Exit                    ; Quit if new buffer starts
                                        ; outside seg 40h
           cmp  ax, 0ff80h
           ja   Exit                    ; ..Or if it ends outside seg 40h.
           cli                          ; Disable interrupts whilst moving buffer
           mov  ds:[80h], ax            ; Start of new buffer
           mov  ds:[1ah], ax            ; Set read pointer to start of buffer
           mov  ds:[1ch], ax            ; Set write pointer to start of buffer
           add  ax, 80*2                ; Adjust for 80 character (word) buffer
           mov  ds:[82h], ax            ; End of new buffer
           sti                          ; Enable interrupts now move is complete
           mov  word ptr es:[bx+14], offset Init  ; Save pointer to start
                                        ; of free memory in Request Header.
           mov  dx, offset Message
           mov  ax, cs
           mov  ds, ax
           mov  ah, 9                   ; N.B. Only DOS functions 1 to 0Ch and 30h
           int  21h                     ; are available at this stage.
           jmp  Exit                    ; Return to DOS
Message    db   13, 10
           db   13, 10, 'KBBUFDEV v1.1   (c) 1989 Lemming Enterprises Ltd'
           db   13, 10, '$'

code       ends
           end  Start
```

Figure 1 – Listing of KBBUFDEV.ASM

The .EXE Disks – Now We Are 3

There are now 3 .EXE Disks, each packed with a megabyte of code and data especially of interest to PC software developers.

The first .EXE Disk was launched in mid 1989 and proved very popular with readers. You asked for more, so here they are. Volumes 2 and 3 are now available, and contain a variety of code listings, executable files and data. The disks are available in 3.5 inch or 5.25 inch varieties, and the price is still £8 each, fully inclusive of VAT and P&P.

The 3.5 inch version is formatted to 720K, and is readable in a 1.44 MB drive. Please note that the 5.25 inch version is a 1.2 MB high-density disk, and will not work in a 360 K drive. Unfortunately, it's just not possible to fit all the files on a 360 K disk. If you don't have access to anything but a 360 K drive, call the office and we'll supply a pair of 360 K disks, but delivery on these will almost certainly take longer than the 7 days in which we normally turn around orders.

You'll need a hard disk with about 2 MB free, in order to copy and unpack the files. Full instructions and documentation are provided on the disk in a READ.ME file. Some of the files on the disks are Public Domain or Shareware utilities, that have been obtained from various databases and online systems. All have been checked with a specialised virus detector and found to be clean.

To order any of the disks, please call the office on 01 994 6477, extension 2359 or 2393, and quote your credit card number. Alternatively, you can send a cheque to us at 10 Barley Mow Passage, London W4 4PH. Please mark your envelope "DISK3" or "DISK5", according to the size that you want, so that it can go straight to the correct department. [EXE]

VOLUME 1

ARCF	ARC	7785	A utility to search one or more ARC files for a text string.
ASMSTRUC	ARC	15360	Macros for Borland's TASM assembler that allow use of if, endif, do while and other structured programming commands in a .ASM file. This version confuses MASM 5, but can be fixed by replacing all ocurrences of &endm with endm.
BACKISS	ARC	12541	ASCII file with brief contents list of all issues of .EXE, up to March 1989 issue.
CIX	DOC	1828	An explanation of Cix, for all who asked.
CLOCK	ARC	7009	Listings from feature on writing an interrupt handler in Microsoft C. From .EXE, Nov 88.
CMOSEDIT	BAS	490	CMOS RAM viewer/editor, from .EXE, Mar 89.
CODEVIEW	ARC	11647	Listings to accompany CodeView articles in Feb and March 1989 issues
DDC	ARC	5632	Disable DOS Commands. Listing from .EXE, Feb 89.
DOS4CURE	ARC	1379	Fixes pop-ups that don't work with DOS 4. See .EXE, Sep 88 (page 62) for explanation.
EXPERT	ARC	4054	Expert system demo in GW-BASIC, from .EXE Dec 88.
GESTALT	ARC	9952	String matching algorithm from .EXE Oct 88.
GVA	COM	264	Get Vector Address. Type GVA xx, where xx is a hex number between 00 and FF, and this program reports the address of the handler for the selected interrupt.
HISTORY	COM	1510	A utility by Robert Schifreen for logging, to a file, all commands typed at DOS prompt. Useful security measure.
INTER988	ARC	166912	350K of useful data that explains all known DOS interrupts, including CD-ROM and MS Mouse calls. As seen in .EXE, Dec 88, page 60.
INTTRAP	ARC	3072	Resident interrupt trapper. Prints values of all registers whenever specified INT is called. As listed in .EXE, Sep 88, page 62.
KEYCODES	ARC	4736	Listing from keyboard article, EXE, Dec 88
LIST	ARC	8249	A wonderfully useful file viewer for ASCII files. Type LIST <filename> then ? for help.
MIXED	ARC	9257	Listings from mixed language programming article in .EXE, Dec 88.
NLOFF	COM	12	A very short utility to turn off NumLock. Useful when placed in your AUTOEXEC.BAT file
OS2LOGIC	ARC	48128	A comms program for OS/2. Works like PROCOMM.
OVERCL	ARC	25661	A demo of OverLay, a program to link in to Clipper that lets you shell to another program after saving most of your current application to disk. Lets you free up all but 10K of RAM before shelling. The full program is sold in the UK by In Touch, who can be reached on 0222 882334
PCX	ARC	29163	Information (plus C code) on decoding .PCX files.
PKXARC	COM	12242	File unpacker
PKARC	ARC	16623	Create archives. The other half of PKXARC. See also ARCF.ARC.
PRTSC	ARC	3559	A resident program that is capable of de-installing itself. See .EXE, Aug 88.
RMAP	ARC	4864	A utility that displays a list of the resident programs currently installed, plus their lengths, their names, and the interrupts they trap.
SNAP31	ARC	208215	SNAP! v3.1 dBASE documentor reviewed in Mar 89.
SNOOP	ARC	36942	Utility, with full MASM source, to display various things about your machine, including list of installed device drivers.
STICK	ARC	3110	New character set for EGA. MASM version of the C listing in June 88.
STYLE	ARC	6123	The .EXE Style Guide, for potential writers.
STYPE	ARC	3072	Smooth typer. A smooth-scroll version of the TYPE command, for EGA screens only
WHEREIS	COM	512	Type WHEREIS <filename>, and this utility will look all through yoour hard disk for it.
WINAPP	ARC	23507	A windows application, by Jay Chapman, that he developed in .EXE starting in late 1987.
XRAY	ARC	4509	A resident debugger that shows you a window on a chunk of RAM, updated in real time

VOLUME 2

EASYCASE	ZIP	251497	The Shareware CASE tool for the PC. Supports a number of graphics cards including EGA and VGA. Includes a large (250K) documentation file which explains exactly how to use the program.
PATTERN	ZIP	1899	Pattern matching algorithm as a BASIC program, as listed in EXE in August 1989
DEVADD	ZIP	6618	Assembly language source code, plus COM executable file, for a program to install DOS device drivers after the machine has booted. That this program needs DOS version 3.30, and minor modifications are required for use with ANY OTHER version.
CXL50	ZIP	257255	CXL version 5. This is a superb text-based windowing and menu package for use with Microsoft or Turbo C. Includes a demo program that shows the capabilities, in both source and executable forms. Registering this Shareware product brings the full source code.
BARCODES	ZIP	5089	Turbo Pascal routines to print Code 39 and EAN 13 bar codes, as printed in EXE Magazine in July 1989
PKZF10	EXE	26298	ZipFind, a search program that scans all files on a disk for a given string. Also looks in ZIP files.
PK2102	EXE	135808	Version 1.02 of PKZIP, PKUNZIP and other utilities. This superb compression tool reduces the size of executable files by 50%, and often shrinks dBASE files to 20% of their original size. Very useful when backing up a hard disk, or when transferring data over telecomms links. ZIPping your software before distributing it can save you 50% in disk costs
NODOS13	ZIP	32952	A utility to create a self-booting disk that doesn't require MS-DOS. Create a program that doesn't make any DOS calls (BIOS calls are OK), then use NODOS to turn it into a self-booting disk. Easy to give copies away, and you don't need an MS-DOS licence.
DIALOG	ZIP	4983	OS/2 PM file selection routines from .EXE, July 1989.

VOLUME 3

INTER589	ZIP	156786	The Interrupt listing, as featured on EXE Disk 1, now up to 1989 version 5, rather than the 1988 version on Disk 1. This version unzips to over 500 KB of documentation on all known MS-DOS and BIOS interrupts, including many undocumented goodies. Highly recommended, and easier to search through than a large manual. This version includes lots on DOS 4, plus DOS extensions such as mouse calls.
FSE	ZIP	13589	Full Screen Editor. GWBASIC lets you move around the screen, put the cursor over a line and press RETURN to re-execute commands that are still on the screen. FSE is a .COM file (with MASM source) that lets you do the same from an MS-DOS prompt screen
BLIP	ZIP	1453	Two MASM routines which are useful for debugging purposes, as printed in EXE in May 1989. BLIP makes a beep without using any DOS or BIOS calls (ie it uses direct hardware calls), so it's safe to call from TSRs or device drivers in order to help follow flow of execution. PRINTREGS prints the contents of all the machine's registers whenever it's called
SOFTLICE	ZIP	3882	A sample software licence agreement, in ASCII format, as mentioned in John Mawhood's .EXE article in September 1989.
MMMMOUSE	ZIP	2284	A utility (COM and .ASM included) to add auto-repeat to the Microsoft mouse buttons. Useful in some applications (especially games!), and as an example of how to intercept mouse calls and pass back values to a program.
LIST72A	ZIP	64571	Version 7 2a of LIST, a Shareware program for listing files in ASCII or Hex format. Faster than loading a word processor
KEYFAKE	ARC	6453	A utility to stuff a string into the keyboard buffer before running a program. Useful, for example, if you want an application to automatically answer its startup questions as soon as it loads.
LIM4SPEC	ZIP	69286	Official, from Intel, the complete specification of the LIM 4.0 expanded memory system. This 400 KB document (when unzipped) provides full details of how to access expanded memory.
PKZF10	EXE	26298	ZipFind. Search disks, including ZIP files, for a string
PKZ102	EXE	135808	Version 1.02 of PKZIP, PKUNZIP and associated utilities. The system will compress files to around half their size. .DBF files typically go down by 60%.
TESS	ZIP	194251	TesSeRact, libraries and documentation to aid in writing terminate-and-stay-resident programs under MS-DOS. As descibed in EXE in March 1989
CIX	ZIP	1610	A brief description of the Cix conferencing system.
OS2POST	ZIP	22764	The text of the OS/2 poster, as given to subscribers to EXE with the August 1989 issue.
TXTUTILS	ZIP	4237	Two small utilities to process ASCII text files. One turns a text file into an executable file. When run, the program displays the text, and lets the user scroll up and down, and print the current page. The other program turns a text file into a pop-up version of the above so that, at the press of a hotkey, the text is on the screen to read
RFC804	ZIP	5459	The specification of Huffman Coding, as used in the international standards for Fax machines
SPLOT	ZIP	1079	Structured Plot C code, from .EXE, July 1989.
CLBEEP	ZIP	1858	A TSR utility that makes your PC beep whenever Caps Lock is hit.

Develop Microsoft® Windows Applications Fast!

The Windows Advantage

You've selected Microsoft Windows as your graphical user interface. Windows applications are easier for end-users to learn and use. It is **the** standard for PCs.

Now you want Windows applications completed as fast as possible. But it takes time for your programmers to get up to speed on the intricacies of Windows development.

Why The Delay For Windows Applications?

Programming for the Windows environment is very different from character-based programming. Just learning the 450 Windows function calls can take months. Getting used to its event-driven system and memory management takes additional time.

Programmers who are new to Windows development are looking for support to help them produce high-quality Windows programs quickly and efficiently.

3 Ways To Speed Up Windows Application Development

Now there are specific ways to speed up applications development under Windows.

- **Code reduction.** There is a fast, effective way to develop Windows applications. A way that cuts the amount of code needed. A way that by-passes much of the complexity of Windows programming. A way that doesn't compromise performance.
- **Short learning curve.** There is an easy, convenient way to get your entire programming staff up to speed with Windows within weeks.
- **From prototype to complete application.** There is a way to quickly develop prototypes of Windows applications in days, not months. Then, turn those into completed applications quickly.

1 Product, 3 Solutions

There is **one** product which brings **all three** solutions to your Windows programmers. It's called **Actor®**

Actor is a proven product. It's now being used in thousands of software development groups world-wide.

Actor Does Windows

Actor gives your programmers a fast, effective way to develop Windows applications. **Actor's** complete development environment allows you to produce everything from prototypes through stand-alone applications. Programs developed in **Actor** are fast and support all Windows features.

Actor offers interactive source code debugging and testing—greatly reducing development time. Your programmers benefit from high code reusability because **Actor** is an object-oriented programming language. This allows them to build on existing code, eliminating duplication of effort. They can even dynamically link to existing C code.

Actor also provides one more essential: fast, responsive support from NEOW - The Windows Specialists, UK experts in Actor and Windows programming.

If you are programming applications for Windows, you should be using **Actor**. Users report that **Actor** cuts Windows programming time by at least 50%.

Act Now!

See for yourself how **Actor** can help your Windows programmers! The first step is to order our free booklet, **For Faster, Better, Smaller, Smarter Windows Programming.** Examine detailed information about using **Actor,** complete with actual source code examples.

Call us at 06286 68334. We'll send you the booklet and complete details about **Actor.**

Take 60 seconds now to save months of development time in the future.

NEOW **- The Windows Specialists**

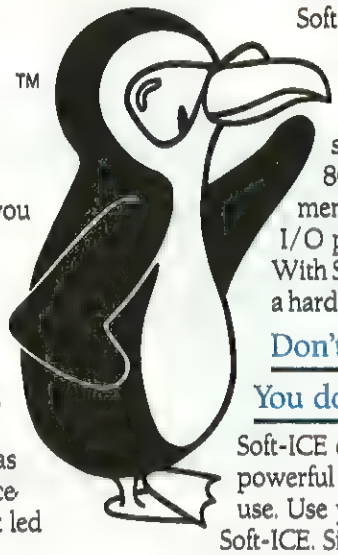
Unit 12, Progress Business Centre,
Whittle Parkway, Slough. SL1 6DQ.
Telephone: 06286 68334. Fax: 06286 65722

CIRCLE NO. 753

FINALLY. A debugging tool tough enough to handle the DOS Nasties.

New Version 2.0

Soft-ICE™



Nasty over-write? No sweat!

Soft-ICE memory range break points help you track down memory over-write problems whether you are doing the over-writing or another program is over-writing you.

Hung program? No problem!

When the system hangs, you now have hope. With Soft-ICE you can break out of hung programs no matter how bad the system has been trashed. And with Soft-ICE's back trace ranges you can re-play the instructions that led up to the crash.

Program too large? Not with Soft-ICE!

Soft-ICE runs entirely in extended memory. This means you can debug even the largest DOS programs. And since your program runs at the same address whether Soft-ICE is loaded or not you can find those subtle bugs that change when the starting address of your code changes.

System debugging? Soft-ICE is a natural!

Soft-ICE is ideal for full source level debugging of TSRs, interrupt service routines, self booting programs, DOS loadable device drivers, real-time kernels, non-DOS O/Ss and ROMs. Soft-ICE can even debug within DOS & BIOS.

How Soft-ICE Works

Soft-ICE uses the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment, while Soft-ICE runs safely in protected mode. Soft-ICE uses the 80386 to provide real-time break points on memory locations, memory ranges, execution, I/O ports, hardware & software interrupts. With Soft-ICE you get all the speed and power of a hardware-assisted debugger at a software price.

Don't want to switch debuggers?

You don't have to!

Soft-ICE can run stand-alone or it can add its powerful break points to the debugger you already use. Use your favorite debugger until you require Soft-ICE. Simply pop up the Soft-ICE window to set powerful real-time break points. When a break point is reached, your debugger will be activated automatically.

MagicCV with Soft-ICE

Using Soft-ICE with CodeView gives you the features necessary for professional level systems debugging. MagicCV and Soft-ICE can work in concert with CodeView to provide the most powerful debugging platform you will find anywhere.

"These may be the only two products I've seen in the last two or three years that exceeded my wildest expectations for power, compatibility and ease-of-use."

—Paul Mace
Paul Mace Software

Soft-ICE	$386
MagicCV	$199
MagicCV for Windows	$199
Buy Soft-ICE & MagicCV(W)	—Save $86.
Buy MagicCV and MagicCVW	—Save $100.
Buy All 3	—Save $186.

30 day money-back guarantee
Visa, MasterCard and AmEx accepted

BYTE Award of Distinction

New Soft-ICE 2.0 features

- Back Trace Ranges
- Symbolic & Source level debugging
- EMS 4.0 support with special EMS debugging commands
- Windowed user interface

Nu-Mega™ TECHNOLOGIES

CALL TODAY (603) 888-2386
or FAX (603) 888-2465

RUN CODEVIEW IN 8K

MagicCV™

CodeView is a great integrated debugger, but it uses over 200K of conventional memory. MagicCV uses advanced features of the 80386 to load CodeView and symbols in extended memory. This allows MagicCV to run CodeView in less than 8K of conventional memory on your 80386 PC.

NEW—Version 2.0 includes EMS 4.0 driver.
Attention Windows Developers!
Version available for CVW.

P.O. BOX 7607 ■ NASHUA, NH ■ 03060-7607 ■ USA.

CIRCLE NO. 754

More dBASE IV

Stop tearing your hair out. If you're having problems with dBASE IV, chances are that you'll find the solution among these current hot topics.

Keep Tabs as Tabs

Unlike dBASE III+, tabs sent to an alternate file via SET ALTERNATE ON in dBASE IV are converted to spaces. If you need to create a text file that retains the tab, use the SET PRINT TO FILE command and send the tab character using the ??? command. Type the following into a .PRG file:

```
SET PRINT TO FILE Tabs.TXT
SET PRINT ON
?
??? CHR(9)+'Hi there'
?
SET PRINT OFF
SET PRINT TO
```

Running the above program yields a text file called TABS.TXT with the string 'Hi there' saved at the first tab position.

CALL() parameter passing

The CALL command and the CALL( ) function are used to execute binary (.BIN) program modules. Once LOADed, the CALLed module can return a value which can be used in a dBASE expression. The syntax of the CALL( ) function is:

```
CALL( ‹expC› , ‹memvar name› )
```

where ‹expC› is the name of the binary module being called, and ‹memvar name› can be up to seven arguments passed to the module. The CALL( ) function returns a value from the first argument only.

Both the CALL command and the CALL( ) function allow the use of up to seven arguments, whereas dBASE III+ allowed only one parameter to be passed with the CALL command. Before executing the module, each argument is checked for validity. The address of the ASCII representation of the first argument, if it exists, is passed in DS:BX. Additionally, ES:DI will point to a 28 byte block of seven 4 byte pointers. Each of these points to the ASCII representation of an argument. Any unsupplied arguments point to a null string. Lastly, the CX register is loaded with the number of arguments passed.

The following dBASE III+ code:

```
LOAD Soundex
USE Namefile
DO WHILE .NOT. EOF( )
  Temp = Username
  CALL Soundex WITH Temp
  REPLACE Code WITH Temp
  SKIP
ENDDO
```

can be replaced with just a single `REPLACE ALL Code WITH CALL("Soundex",Username)` command in dBASE IV.

Renaming A .DBO File

If a compiled .DBO file is renamed, the resultant file cannot be run. Attempting to do so will return a 'Procedure not found' error.

Every program file becomes its own main procedure file. When a file is compiled, its name is embedded into the resultant .DBO. Simply renaming a file will not change the references to the original filename in the .DBO file. When the newly named file calls itself as a main procedure, this name is not the same as the filename of the main procedure compiled into the .DBO, and the error is returned. To get around the problem, rename the .PRG file and recompile.

Differing INKEY() Values

Some keystroke values returned for INKEY() have changed from dBASE III+, and the wrong values have been documented on page 4-55 of the Language Reference manual. The table below contains the values returned by both products:

Key	dBASE IV	dBASE III+
Home	26	1
End	2	6
Ctrl-RtArrow	6	2

On the same page, Ctrl-F is not Ctrl-End but it is actually just the End key.

Header Structure

The dBASE IV header structure has been documented incorrectly on page E-2 of the Language Reference manual. The byte 0 information should be as follows:

Bits 0,1,2	indicate the version number
Bit 3	indicates the presence of a dBASE IV memo file
Bits 4,5,6	indicate the presence of a SQL table
Bit 7	indicates the presence of dBASE III+/dBASE IV memo file

All other information on Tables E-1 and E-2 is correct.

@..GET..PICTURE @M

When @...GET...PICTURE...@M is issued, subsequent PICTURE templates are ignored. Rather than displaying the respective values in the specified PICTURE formats, the default lengths and formats are displayed. This will also affect reports and labels that are run subsequent to @...GET...PICTURE...@M. In the following example, 'dummy' is displayed, rather than 'd'.

```
items = space(10)
mvar2 = "dummy"
@ 5,5 GET mvar1 PICTURE
      "@M item1,item2,item3"
@ 6,5 SAY mvar2 PICTURE "X"
READ
```

To avoid this problem, do the following. At the beginning of your program, initialize a variable to a null string. Then, GET that memory variable on the same line as the @...GET...PICTURE "@M", but with the column coordinate at the right edge of the screen. For example:

```
junk = ""
...
@5,5 GET mvar1 PICTURE
     "@M item1,item2,item3"
@5,75 GET junk
...
```

SEEK()

If the SEEK( ) function is issued where the function's lookup expression uses a field in the current work area, and the lookup is performed on a file in an unselected work area, corruption can occur in the currently selected database.

If the SEEK( ) is performed with a character field, the field immediately following that field in the file structure will appear to be blank. If no changes are made to the record in the selected work area, no permanent corruption occurs. However, EDITing the record will permanently corrupt the following field's value.

If the SEEK( ) is performed with a numeric field, then DISPLAY STRUCTURE will reveal corruption in the width of the numeric field. If this damage occurs, and the damaged record is edited, the database will become permanently corrupted. The same is true if the SEEK( ) function is issued from within a .FRG, .LBG or .QBE file.

The work-around is as follows. If SEEK( ) is used in a .PRG, store the field to a memory variable and use the memory variable as SEEK's expression parameter. If SEEK( ) is used in a .FRG, .LBG or .QBE, create a calculated field whose expression is the field's name, and use this calculated field as SEEK's expression parameter.

DBSETUP

If a maths co-processor is installed, the DBSETUP program displays a blank entry in the co-processor field, rather than displaying a *YES*. dBASE IV does recognize and make use of a maths co-processor to increase performance, even though this display error would seem to indicate otherwise.

Additionally, the same display error occurs in the DMA controller field of the DBSETUP system configuration screen when using a DMA controller.

DBSETUP and Colours

If you use DBSETUP to change the colour configuration, the new colour settings are not stored in CONFIG.DB and are, therefore, lost. There are 2 possible work-arounds for this. First, you can edit CONFIG.DB manually. An alternative method is to delete all of the COLOR OF commands from the CONFIG.DB file. If you do this, DBSETUP *will* save new colour settings to the file.

Fonts in dBASE IV

The dBASE IV system allows up to four printers and up to five fonts for each of those printers that are defined. However, the way commands are sorted in CONFIG.DB, combined with errors in the documentation, may hinder the implementation of user-defined fonts.

Fonts can be defined by either using the DBSETUP utility or by directly editing the CONFIG.DB file. To set up fonts in DBSETUP, type DBSETUP and select the CONFIG.DB menu, then select the appropriate choice to either create or modify the file. Now, select Output:Printer:Font. This brings up the Font menu which allows the entering of descriptive text as well as the starting and ending control codes. Consult your printer manual for available fonts and their control codes. Once the desired fonts have been entered, save the CONFIG.DB file and exit the DBSETUP utility. After setting up the fonts, use a text editor (or the dBASE program editor), to modify the CONFIG.DB file and move the PDRIVER statement after all of the PRINTER statements.

Alternatively, the font definitions can be entered into the CONFIG.DB file using the PRINTER statement. The PRINTER statement is described in the Language Reference manual on page 6-7. However, the example given is incorrect. The correct syntax is:

```
PRINTER 1 FONT 1 = (Esc)(8U,
  (Esc)(#@ NAME "Roman-8
  Symbol Set"
```

There should not be a comma after the ending control code. The = sign between the keyword NAME and the descriptive text is optional. The font will appear on the Words:Style menu as '1. Roman-8 Symbol Set'.

Once you have set up all this, there's still a problem. It is not necessarily obvious which printer is the 'active' one. Initially, the default printer driver is determined by the PDRIVER statement in the CONFIG.DB file. The only way to change the default printer driver to a different one is by assigning the desired driver to the -pdriver system memory variable. So, to use fonts for a printer that is not designated as the default in the CONFIG.DB file, a -pdriver command must be issued at the dot prompt to change to the appropriate driver. An example of this command would be -pdriver = "GENERIC.PR2". When this command is successfully executed at the dot prompt, the message 'Printer driver installed' will appear and the font definitions in CONFIG.DB that correspond to that printer driver will load.

The Report design screen assumes that the default driver is GENERIC.PR2. So, initially it appears that no fonts are available on the Words:Style menu. To access the fonts, the Print:Destination:Printer Model option must be changed; the fonts will then appear on the Words:Style menu.

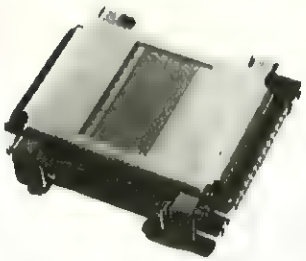
Note that changing the Print:Destination:Printer Model option does not load the corresponding printer driver; it merely allows report formats using these fonts to be created. To print the report, the appropriate -pdriver must be activated, either by the PDRIVER statement in CONFIG.DB or setting the variable -pdriver at the dot prompt. Finally, the desired fonts should be accessible and printable.

No .BAK Files

Issuing the command MODIFY STRUCTURE from the dot prompt, or using Shift-F2 from the command center, while highlighting a database file, allows modifications to be made to the structure of the .DBF file. dBASE III+ automatically creates a backup file with the extension .BAK that is a copy of the original .DBF file, but dBASE IV does not create this backup. In order to preserve data in the case of a mishap or an unsuccessful modification to the existing structure, use the COPY command to produce a duplicate, temporary file prior to modifying any .DBF file. .EXE

Many thanks to Julie Cox and her team at Ashton-Tate for scanning their telephone logs to provide these work-arounds. If you're hungry for more, there's a patch disk available, which is currently up to version 1.07. This also contains a dBASE IV mouse driver, new printer drivers and a new set of .GEN files for use with the Control Center. The disk can be obtained, free of charge, from Ashton Tate's technical support department at Oaklands, 1 Bath Road, Maidenhead, Berks, SL4 4UH.

SOFTLoK™ & SOFTLoK PLUS™ software protection systems



- For IBM PC, PS/2 and compatibles
- Installed in seconds in the parallel printer port
- Assembler & high level source code provided
- Fast response
- Transparent to printer
- Small physical size

SOFTLoK

SOFTLoK units contain a code that can be read and verified by routines provided in assembler and a variety of high level languages.

Price: 10-99 units £12.00, 100+ units £8.00
Evaluation kit £20.00

SOFTLoK PLUS

SOFTLoK PLUS units contain a small area of non-volatile memory protected from unauthorised reading or writing by a password. Both the memory area and the password can be changed at any time by the routines provided in assembler & several high level languages. As the memory can even be altered in the field it can contain counters & expiry dates as well as serial numbers etc.

Price: 10-49 units £40.00, 50+ units £35.00
Evaluation kit £40.00

SOFTLoK International Limited
Tel: (0254) 77220 Fax: (0254) 705956

CIRCLE NO. 755

S-GKS portable graphics toolkit

- available on a wide range of systems from PC's to mainframes
- support for the popular workstations from SUN, Apollo, IBM, HP, Sony, DEC
- Fortran and 'C' language support
- X-Windows driver
- CGM (Computer Graphics Metafile) support
- Training courses run for GKS programmers

B-GUL graphics toolkit

- Business & technical graphics toolkit layered on top of GKS for efficiency and portability
- Comprehensive charting facilities including;
  Bar-charts, Pie-charts, XY-plots
  Word-charts, Tables, Keys, Hi-lo-charts
- Pasteboard facility for composite plots
- Powerful data scaling & axis facilities
- FORTRAN and 'C' language support

Scientific Software Limited
Rose Industrial Estate
54 Marlow Bottom
Marlow, Bucks SL7 3ND
Tel : 0628 890011

CIRCLE NO. 757

System Science

C COMPILERS

Microsoft C ver 5.1	£235.00	Turbo C	£75.00
Microsoft Quick C2	£85.00	Turbo C professional	£155.00
Aztec C86 Comm	£245.00	Watcom C ver 7.00	£235.00
Aztec C86 Dev	£155.00	Lattice C (new ver 6.0)	£185.00
Zortech C++ Ver 2.0	£125.00	Glockenspiel Common View	£375.00
Zortech C++ Prof Dev	£275.00	Quick C with Quick ASM	£129.00

C GRAPHICS

Halo-86 (also Qb, Fort, Pas)	£225.00	Enhanced Graphics T'kit	£175.00
Metawindows-no royalties	£185.00	Metawindows Plus	£245.00
Ingraf-C-source, no royalties	£195.00	Essential Graphics s'rce, nr	£155.00
PCX programmers Toolkit	£135.00	MS Windows SDK	£295.00

C DATAFILE

Btrieve (also QB, Pas, Fort etc)	£169.00	Ctree—source	£245.00
Btrieve/N -network	£395.00	Fabs Plus	£125.00
XQL addon for Btrieve	£525.00	Code BASE 4	£195.00
Xtrieve option	£395.00	CQL for Ctree or Btrieve	£250.00

C SCREENS

Vermont Views (new)	£275.00	Vitamin C	£145.00
Panel Plus II	£275.00	Greenleaf Datawindows	£245.00
C Worthy	£call	Escape with Look & Feel	£345.00

C COMMS AND OTHERS

Greenleaf Comms	£195.00	Essential Comms s'rce	£125.00
Blaise C Asynch Manager	£125.00	Pforce—comms, dbase, scrns	£225.00
Blaise Turbo C Tools- new ver	£95.00	Blaise C Tools Plus	£89.00
Greenleaf Functions	£149.00	Greenleaf Super Funcs	£195.00

PROGRAMMERS TOOLS

Microsoft Assembler 5.1	£90.00	Turbo Assembler/Debugger	£99.00
Brief Editor (DOS or OS/2)	£145.00	Epsilon Editor (DOS/OS/2)	£145.00
Norton Editor	£45.00	Vedit Plus new ver	£99.00
Plink 86 Plus	£395.00	RT Link	£155.00
PolyMake	£125.00	PVCS Corporate	£375.00
MKS Toolkit	£175.00	Vq matchmaker & editor	£175.00

CROSS DEVELOPMENT

2500AD Cross Assemblers	£135.00	2500AD Simulators	£115.00
Avocet Cross ASM	£295.00	Avocet Simulators	£395.00
IAR Cross Assemblers	£185.00	Avocet C Cross	£695.00
IAR C Cross Z80, 6301 etc	£995.00	Introl C Cr. 6809, 6301, 68000	£1445.00
Aztec C Cross	£275.00	LMI Forth Metacompiler	£995.00

FORTRAN ETC

Microsoft Fortran with CodeView	£235.00	RM Fortran with Forte	£485.00
Lahey F77L (new version)	£395.00	Pro-Fortran 77	£295.00
Lahey F77/32EM for 386s (new version)	£795.00	Ingraf Fortran source	£195.00
Metawindows	£185.00	Enhanced Graphics T'kit s'rce	£175.00
GSS GKS Graphics	£385.00	FortLib source	£95.00

PASCAL ETC

Turbo Pascal 5.5	£79.00	Blaise Power Tools Plus	£99.00
Turbo Pascal Professional	£155.00	Blaise Turbo Asynch	£125.00
TurboPower Prof	£89.00	Microsoft Pascal 4.0	£185.00
Fabs Plus Datafile	£125.00	many more for Turbo Pascal	£call

BASIC ETC

Microsoft QuickBasic ver 4.5	£85.00	Quick Comm	£95.00
Microsoft Basic 6.0 Compikler	£175.00	Crecent Quick Pak Prof	£115.00
Btrieve	£169.00	Fabs Plus Datafile	£125.00
Blaise Power Screen	£95.00	many more	£call

OTHER LANGUAGES

JPI Modula-2	£59.00	Logitech Modula-2 Develop	£175.00
MuLisp 87 Interpreter	£175.00	MuLisp-87 Compiler & Interp	£225.00
Smalltalk V	£79.00	Smalltalk 286	£129.00
PC Forth by LMI	£95.00	UR/Forth by LMI	£225.00
Microsoft Cobol II	£445.00	PC Logo	£75.00

UTILITIES & OTHERS

Uniform/PC read CP/M disks	£62.00	Dan Bricklins Demo II	£175.00
Spin Rite II	£69.00	Norton Utilities Advanced	£75.00
Unidos with 8MHz Z80	£125.00	Vcache by Golden Bow	£49.00
Compaticad II	£95.00	Vfeature Deluxe	£85.00
dBrief addon for Brief	£79.00	Tom Rettigs Library	£75.00
Above disc Lim Dr	£69.00	Math CAD 2.5	£345.00
Microstat II	£295.00	Derive—computer Algebra	£115.00

386 AND OS/2

386Max Prof	£95.00	VM/386	£115.00
DesqView 386	£115.00	Xenix 386—new price list	£call
OS/2 1.1 Stand Ed (IBM)	£call	OS/2 PM Toolskit	£295.00
Many OS/2 Libs & utils	£call	many 386 Utils	£call
Multi-boot (OS/2)	£45.00	Win Pro/PM	£250.00
Smalltalk/PM	£325.00	Multiscope Debugger	£195.00

All prices are exclusive of VAT but do include UK delivery.

A sample only of our extensive range. Please enquire for unlisted items

Prices are subject to change — Please confirm

3-5 Cynthia Street, London N1 9JF
Tel: (01) 833-1022,
Fax: (01) 837-6411

BARCLAYCARD VISA

CIRCLE NO. 756

PharLap's 386 Assembler Ensemble

The 386 has the ability to do a lot more than MS-DOS normally allows it to do. Vik Olliver looks at the PharLap tools, which give you access to the whole of the CPU.

The 386 is a very powerful beast. It addresses up to 4 Gbytes of RAM, has 32 bit registers, and has done away with that awful kludge called segmentation. What do most people do with it? They run it either as a fast disk server, or run 640K MS-DOS on it. With MS-DOS's addressing capability limited to 640K, all that extra RAM goes to make a RAM disk, or a disk cache, or runs multiple copies of DOS under something like Desqview. It's just too much hard work to re-write programs to use all the 386's capabilities.

Enter PharLap. Their 386 DOS extender allows programs to use traditional DOS calls in a 386 protected mode environment, with access to all the memory and the rest of the 386's goodies. Without this kit, there's little you can do. You can, of course, use some of the 386's extra opcodes (like those that allow pushing and popping of multiple registers), but the 32-bit registers are beyond reach.

Of course, you need to re-write your code to run in protected mode, so PharLap also make an assembler that is capable of assembling 386 code without too much bother. Not much to re-learn, as the MS-DOS API stays the same.

Having assembled your code, it is traditional to link it into some form of executable file. To do this, you need a linker and, surprise, surprise, PharLap have one of those as well. Linkers on a 386 system, though, have to cope with much more than .EXE files: More memory, potentially more segments, run time restrictions on segments and so forth. This calls for a new executable file format, and PharLap have invented one, which they call the .EXP file.

Once you have a .EXP file, you'll want to debug it. With the DOS extender comes a 386 version of DEBUG ('MINIBUG'), which is not what you want to use for debugging programs that make full use of the 386's facilities. PharLap sell a much more elaborate debugger, and this has symbolic support.

So, having skimmed the surface, let's go into more detail about each of these components.

The DOS Extender

The concept behind a DOS extender is simple enough; switch between real and protected mode, execute DOS calls, and look after the memory. In practice, there are problems that make the extender worthwhile. The extender can cope with up to 4 Gbytes of memory, compared with DOS's 640K. So the extender has to cope with buffers for DOS calls being out of the standard address space. Registers are also 32 bits wide now (AX is extended to EAX, in the same way that AL is extended to AX, and so on), so handle-oriented reads and writes of over 64K are now possible with one call. Memory is managed in 4K pages, rather than in 16-byte paragraphs as with DOS. Each page allocated is given its own LDT (local descriptor table) by which it is known.

Interrupts on a 386 are a problem. If an interrupt (say, from the system clock) happens when the 386 is in protected mode, the processor has to switch into real mode to execute the interrupt handler. This all happens automatically. The DOS extender allows a programmer to patch real and protected mode interrupts to different locations, and also to patch an interrupt so that the subroutine always gains control in protected mode.

Memory can be allocated from a known address to satisfy the needs of DMA devices, and a descriptor is provided which maps the whole of memory as one linear address space. A nice afterthought is a descriptor that always points to the base of video RAM, even on a VGA. Other descriptors (all with fixed numbers and documented) reference the code, data, stack, PSP and, by some magic, a Weitek co-processor if it is fitted. As for memory, the extender uses about 100K of conventional memory, and an additional 100K of conventional memory is needed for each 386 program which is run by the original.

Distributing Software

Once you have written and compiled your program, the extender is the program that loads and executes .EXP files. For applications that you intend to give away or sell, PharLap will sell you a licence for around &1000. This special licensed version of the extender is then bolted seamlessly to the start of your code, which results in a single .EXE file, and the end user is none the wiser as to what is going on other than that he has to run it on a 386.

Assembler

The assembler is very MASM-like, as most assemblers seem to be. Some of the strange parts of MASM are missing, such as the inability to do near jumps between groups. Like MASM, PharLap's assembler loves inserting free NOP instructions at strategic points in your code.

One quibble with the assembler is its command line interface. Using SET INCLUDE=C:\INCLUDES no longer works, and the -INCLUDE directive has to be used on the command line. This is all very well, but I use a lot of includes, and command lines are of limited length. The command line interface is also a bit fussy about whether or not you have a terminating backslash on your paths.

Another niggle is the way that some instructions (IRET and PUSHA) are not converted into the 32 bit equivalent, even though the code is written with a USE32 directive. You have to specify IRETD or PUSHAD for the 32 bit instructions yourself. Other than that, it was very easy making the transition from MASM to 386ASM.

Linker

While most assemblers can be persuaded to generate some form of 386 code, linking it is a whole new kettle of worms. The obvious difficulties are that segment registers as found on the 8088/86 no longer exist in a 386 that's running in protected mode, and are replaced with descriptors. The programs can also be (in theory, anyway) nearly 4 Gbytes long.

To cope with this, the linker links all the code, data, and stack segments of a .EXP file into one large program (called a 'flat' model), and points descriptors at it. The data descriptor and the stack descriptor are identical to the code segment, but are writable and not executable. For those expecting independent segments for stack and data to help with debugging, or who fancied splitting code into distinct modules, this is a disappointment. It also makes it possible for the code to be overwritten by references to the data segment.

My company's solution to this was to allocate lumps of memory at the start of the program, and remove the data descriptor that could reference the code. Sadly, the only way of referencing the data then was to use structures, which are long-winded to use in the assembler, and cannot be declared as global symbols for the debugger. Using the PharLap linker is no more difficult than using any other. All options go on the command line. Unlike the assembler, though, there is a way of setting your INCLUDE paths in the environment space.

Among its options, the linker will output .EXE (but not for protected mode code) and .REX files, as well as .EXP files. The .REX files are a relocatable executable format used by other manufacturers. The linker can produce several different varieties of hex file with which to blow EPROMs, including Intel and Motorola formats. Packing of .EXE and .EXP files is an inbuilt feature of the linker, so no separate EXEPACK-type utility is required. Likewise, the minimum and maximum sizes of the file are also a function of the linker.

One very handy feature of the linker and .EXP files is that the symbol table can be appended to the .EXP file with a command line option. This is very useful in preventing your symbolic information getting out of sync with the code as you run around the office trying your code out on different machines.

The linker can also be forced to check if the code that is being linked overlays itself. This usually happens when one block of code overflows its allotted space (when making code with a fixed jumpblock in it, for instance), and floods into an area which is ORG'd just below it. The linker supports indirect command files, which can contain both filenames and command line parameters. Comments can be used in command files to avoid confusion.

The Debugger

386DEBUG understands .EXE and .EXP file formats, and superficially resembles SYMDEB in operation. It is, of course, symbolic, and supports symbols, registers and a few mathematical functions in the commands.

In our office at least, this was the least favourite program of the suite. When disassembling short jumps, the debugger only gives the destination as a symbol plus offset, and does not give the real address as SYMDEB does. Absolute values loaded into registers are given, but indirect values are not, so when you see CMP AL,[ESI] you have no idea what ESI is pointing at. As the symbol table is loaded from the tail end of the .EXP file, multiple symbol files are not supported. Tracking down the nearest symbol is much easier, and there is a command for this, as well as one to list all the symbols.

Unlike SYMDEB, it is not possible to assemble code while in the debugger, so making trivial modifications to code requires hand assembly, or assembling and linking the program again. Entering hand-assembled code is fun, as the memory modify command is not interactive, so the address and data must be entered as one command.

Tracing fares a little better, in that not only does it allow a given number of instructions to be traced, but this can be done without output to the screen. As the tracing of 386 programs is done using the 386 debugging hardware, stepping through a ROM is no different to stepping through RAM. Stepping through any kind of reentrant interrupt code usually caused a crash, as did tracing through exception handlers.

The 'single step, skipping calls' command, the P in SYMDEB, also has a 'quiet' trace facility, but tends to stop on the instruction after the first call it hits, though this is now what the manual says should happen.

Breakpoints are implemented using INT 3 when debugging .EXE files, and hardware breakpoints when debugging .EXP files. Introducing your own INT 3 to your code in an attempt to cause a breakpoint causes the debugger to halt with an exception. It then dumps you at the DOS prompt, where you can't do anything about it. This happens for several exceptions, and is extremely annoying.

There are no count or conditional options for breakpoints and, other than breakpoints, no way of stopping code from executing apart from the usual Ctrl-C when a DOS call is being executed. This is only useful when you're using a lot of DOS calls.

Watchpoints are easily implemented on a 386 system, so it's no surprise that DEBUG386 has them. The debugging registers can trap reads and/or writes to a specific memory location, and also whether the operation is byte, word, or doubleword in length. Again, there is no option to break on a specific value, or for a count. This was extremely annoying, and we ended up writing our own debug traps to implement value and count sensitive breakpoints. Register display commands are enhanced, and come in several flavours, giving the basic register set, along with all 386 registers and the global and local descriptor tables.

When your code stops, the debugger tells you how long the code takes to run to the nearest 1/10th of a second, and knows enough not to give times when you have patched the clock. I found this useful when writing speed-critical code, but it did occasionally give very silly answers in the order of 4 million seconds.

Two things PharLap have thought about are that you might want to disassemble 16 bit code while running 32 bit code, say in an interrupt routine or while searching the ROM, and that you might not want to use the console. To this end there are U16 and U32 commands, which you'll use more often than you think, and I/O redirection to a serial port that handles Ctrl-C properly.

Although the debugger has its problems, these can be worked round. It is a heck of a lot cheaper than a 386 ICE, and there don't seem to be many alternatives.

Documentation

The manuals come in the usual ring binders, 9 pages of release notes at the front, a

license agreement that accounts for a large quantity of rainforest and ink, and loading instructions tucked into a flap on the cover.

The debugger instructions were the only ones with a problem: they referred to a directory by the wrong name, but it was obvious what they meant.

Included with our manuals was a copy of Intel's 80386 Programmer's Reference Guide, and a copy of the Intel 80386 System Software Writer's Guide. The debugger and utility instructions come in one manual, assembler and linker in another, and DOS extender and 'MINIBUG' in a third. The DOS extender is provided as part of the ASM/LINK kit.

Instructions in all but the debugger and DOS extender sections are generic, and explain the use of the programs on DOS, VAX/VMS, and UNIX systems. The format of the .EXP file is revealed in the linker manual, so it is possible to write your own loader if you must. Another way we tried was putting a small routine at the end of the code that wrote the entire program out as one long binary file. This can be loaded as an overlay or as a program in its own right by a relatively simple homebrewed loader.

The PharLap manuals are easy to reference, contain more than enough information, and have a usable index to boot. Areas of the manual documenting the more shady aspects of 386 programming are showing definite need of hole reinforcement rings.

There is a fairly large example program supplied, which uses a mouse to demonstrate how to call real mode code from protected mode code and back again. Fortunately, the manual is clear enough, and you don't have to delve through it. The Intel documentation however, has no index, scatters its information throughout the manual, uses inconsistent terms and reminds one of chip documentation in general.

Bits and Pieces

The assembler and linker will run on any DOS-based machine, even a base PC or an aged Amstrad. Versions are also available that cross-assemble on VAX/VMS, and UNIX. The standard linker can run out of symbol table space, and a version running under the DOS extender is provided to prevent this. Of course, it'll only run on a 386 machine.

Hiding in the wings is a configuration program for all of the utilities. It allows you to set the default command line parameters, and more programs could do with utilities like this. PharLap also have a 386 library utility, but this was not tested.

Conclusion

The PharLap suite of programs is an efficient way of expanding a DOS environment to a 386 system without having to use radically different programming techniques. The DOS extender especially should extend the useful life of DOS on 386 machines considerably.

Vik Olliver is a programmer at London-based Alfa Systems Ltd, where he is working on a 386-based programming project using PharLap's tools. Products compatible with PharLap include MetaWare High C; MetaWare Professional PASCAL; NDP C-386; NDP FORTRAN-386; LPI COBOL, BASIC, and FORTRAN 386; LMI FORTH Metacompiler 386 and the Alsys ADA-386 compiler. The PharLap ASM/LINK kit costs £325, and the debugger is an extra £135. UK supplier is System Science (01 833 1022). PharLap are based in Cambridge, Mass., and can be had on 0101 617 661 1510.

Verification, Validation and Testing

With a little thought, it should be possible to drastically reduce the number of bugs in your code. John Bruce has some suggestions.

There's a school of thought that puts the 'industry standard' bug ratio at 50 errors per thousand lines of code (1). If this is the case then, with a little thought, there is no reason why this figure should not be improved by a factor of 100. Doing so would provide increased reliability and reduced maintenance costs, too. In this article, I'll highlight some of the ways in which verification and validation techniques can be used to help achieve fewer errors.

The 'Cleanroom' Method of System Development

The technique (5) is based on non-execution-based program development and independent statistically-based testing. Figure 5 shows that Cleanroom is an integration of formal specification/design, incremental non-execution-based development, and independent statistically-based testing. Mathematical verification replaces debugging and a product of certifiable reliability is the aim.

The development process is a cycle of producing executable increments which are passed on for testing. These software increments accumulate until the system is complete. Next is a Formal Specification, written in a development language such as the Vienna Development Method (VDM), OBJ or Z. The correctness of the program is assured and it is claimed that this verification process avoids the necessity for program testing. After this a structured program is produced. One of the nice features of the cleanroom approach is that the formal specification can be left in the program as non-executable comment statements. It provides an excellent form of documentation. For many years, Pascal programmers have used this technique of constructing a program in pseudo-code-type comment statements, and then adding executable statements in the appropriate places after words. Modula-2 and Ada programmers now do the same.

In the Cleanroom concept, statistical testing is based on MTTF (Mean Time To Failure) and expressed as –

$MTTF = MR^{c}$ for c software changes

where M is initial time to failure and R is the observed effectiveness ratio for improving MTTF with software changes. A technical rationale for this is given in Reference 5.

In common with conventional testing, statistical testing is intended to simulate an operational environment. It is a system of randomly selecting test cases which are based on:

1. *A frequency distribution of inputs (commands and data) to the system, such as the different types of input transactions in a Order/Entry system, also on.*
2. *A frequency distribution of machine states.*
3. *The range of capability of the system as it expands.*

It is said that the Cleanroom approach is no more expensive than conventional methods. Mills et al (1987) gives the example of a 20 KLOC program which had only 53 errors instead of the 1000 expected by the previously stated estimated industry average of 50 errors per 1000 lines of code. Mills (et al) state that a post-delivery rate of less than one error/KLOC is feasible with Cleanroom, as compared with the normal rate of one to 10 errors/KLOC for execution-based production. Also claimed is a productivity of more than 400 lines per man month, largely due to reduced testing time. There is also a very marked improvement in reliability, which cuts maintenance costs dramatically. The technique seems to be a major advance in the fundamental principles of software engineering.

Verification and validation are two sides of the same coin for which Boehm (2) gave us these simplifications :

- Verification – 'Are we building the product right?'
- Validation – 'Are we building the right product?'

In the dictionary sense, to verify is to test the truth, accuracy or correctness of something. In our case this is the assertion 'are we building the product in the right way'. Validation, on the other hand, is the act of establishing that all necessary conditions are fulfilled. This applies to the specification by posing the question about 'the right product'. Confirming, 'is this what the customer asked for?', is perhaps a sharper focus. There can be subtle differences between that answer and what is actually wanted.

Testing

The concept of testing extends to embrace not only program execution with suitable data, but the processes of verification and validation as well. Incidently, none of the methods to be described here requires that the program be executed, therefore, data-based testing is excluded.

There are four broad categories of techniques available for software verification and validation. These vary from quite informal manual methods which do not require any tool, up to very formal mathematical methods.

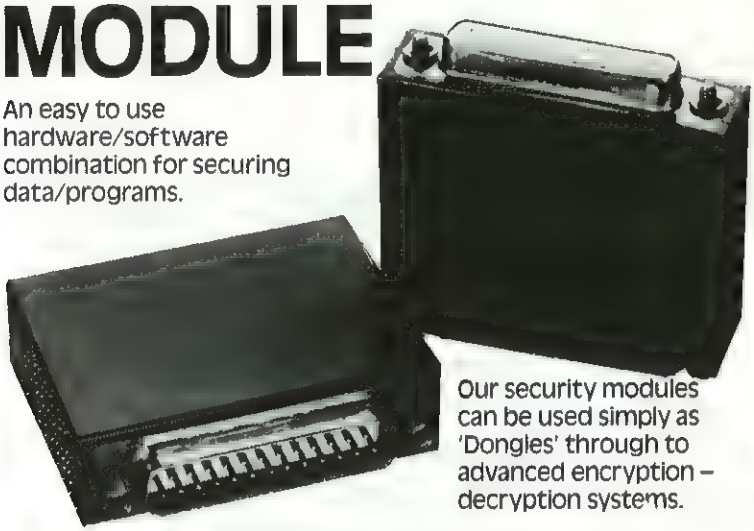
- Manual methods.
- Static analysis.
- Dynamic analysis.
- Functional testing.

Within the first category, four commonly used manual methods are:

- Walk-through.
- Review.
- Inspection.
- Audit.

SOFTWARE SECURITY MODULE

An easy to use hardware/software combination for securing data/programs.



Our security modules can be used simply as 'Dongles' through to advanced encryption – decryption systems.

- ★ For IBM PC family (and compatibles)
- ★ Variants for other systems
- ★ Advanced construction – rugged and compact
- ★ Proven throughout the world
- ★ Transparent operation – user friendly
- ★ Driver software supplied for variety of applications

CTL

Control Telemetry of London
11 Canfield Place, London NW6 3BT
Tel: 01- 328 1155 Fax: 01- 328 9149

CIRCLE NO. 763

EVERLOCK

COPY PROTECTION

Designed for clone compatibility & strength

- Thwarts ALL Bit-copy Software
- Protect any COM/EXE w/o Source changes
- Shut down Debug Tracing & Disassemblers
- Supports all Hard & Floppy Disk formats
- Restrict or allow Network Installs
- Limit program use by Date or # Executes
- Remotely reset Installs, Date or # Executes
- No damaged media or I/O plugs

For IBM or Clones £465.00 + p&p £6.50
FREE INFORMATION & DEMONSTRATION
Limited Use Version £150.00 + p & p £6.50
(Prices exclude VAT)

USER FRIENDLY micro computer systems

33, Edgmond Close, Winyates West,
Redditch, Worcestershire B98 0JG
Tel: 0527 510105 Fax: 0527 20920

COMPANY NAME _____

ADDRESS _____

_____ POSTCODE _____

Please send me _____ copies of Everlock at £ _____

I enclose a cheque / postal order to the value of £ _____

inc VAT and p & p. Send Information: ☐

CIRCLE NO. 761

OREGON C++

KILLS ALL KNOWN BUGS

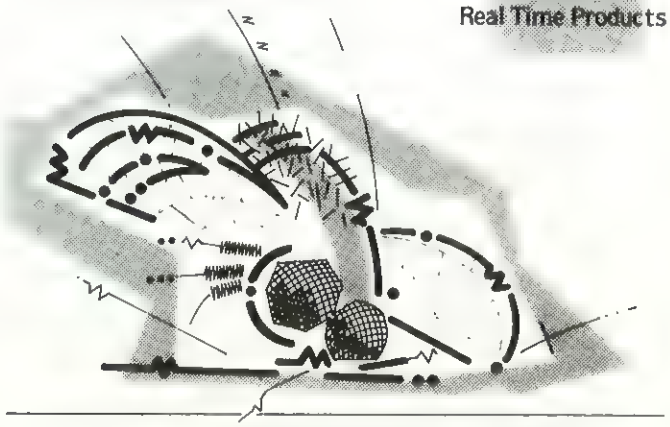
C++ represents a new generation of object oriented programming languages. It provides you with the techniques you need to implement large, maintainable software systems. C++ is also a superset of C, which means that you can introduce these techniques without abandoning your existing investment.

Oregon C++ is a powerful optimising compiler that generates extremely fast, compact code from C++, ANSI C and pre-ANSI C source. It is supplied with the Oregon Debugger, a full function, windows based symbolic C++ debugger. Based on the same optimisation technology that brought you Oregon Pascal and Modula-2, Oregon C++, is available on Vax, Sun, 386/Xenix and other hosts.

TRAINING
▪
C++ & OBJECT ORIENTED DESIGN
▪
CALL 021-236 8070

For more information on C++, or our other language development tools, call 021-236 8070.

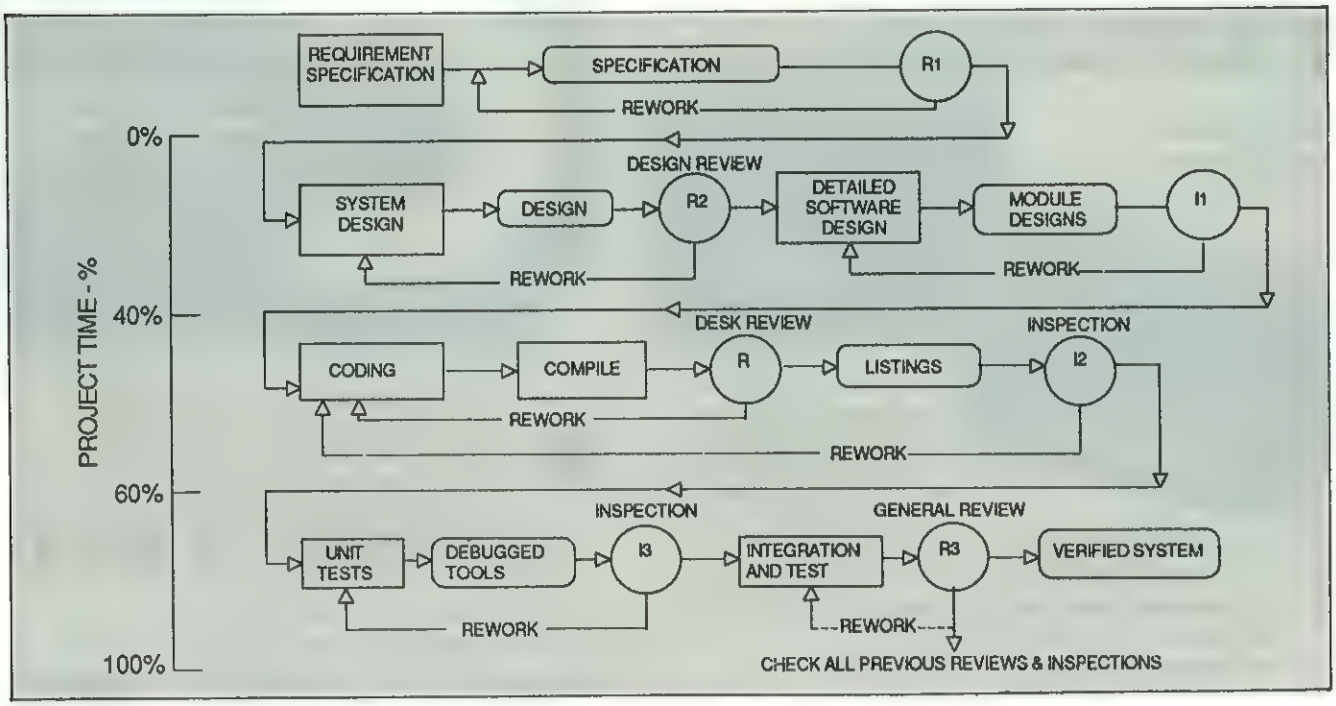
RTP
Real Time Products



[] PASCAL [] C [] MODULA-2 [] C++ [] CROSS & NATIVE []

CIRCLE NO. 762

Figure 1 – Life-Cycle Model Showing Review and Inspection Points



A fifth method, invented by IBM, is beginning to gain favour. It is the Cleanroom technique, so named by comparison with the special facilities provided for hardware manufacture where the entry of defects is denied. (A brief outline is given in the box entitled The Cleanroom Method of System Development.)

Walk-Throughs

Essentially, in a walk-through, someone will talk through the object, explaining in detail how or why he did something, and particularly how it is expected to work. It's like a guided tour. If it ends there, without any formalities, it is a walk-through. When the meeting is extended by having the object introduced first, followed by a walk-through where errors are recorded, and concluding with a summary of findings, then we have a review. The informality of walk-throughs is so variable that no standard approach can be recommended. However, a chairman, secretary and an agenda is desirable. In most cases the purpose of a review is to examine a single object, particularly source code, but systems design, documentation, test plans and so on may also be the subject.

An essential aspect of these reviews is that they are concerned with error detection, not error correction. Consequently, a review meeting may discover many errors on one single occasion, instead of having to wait until individual errors are found by normal debugging techniques. The time saved in this way can significantly reduce system validation effort.

Those attending walk-throughs should be peers of the person whose work is being considered, although this may not be so for code audit, or code inspection. In the psychological sense, it is quite important that the process is non-threatening and does not involve Management. Similarly, the results of inspections should not be used for programmer performance appraisal. Defensive behaviour is then avoided.

It is essential in project management to have a clearly defined sequence of operations in a project, and a life-cycle model can provide this. An example is shown in Figure 1, where the whole project is allocated time on the basis of the software engineering general 'rule of thumb', that is – 40% design and development, 20% coding and 40% testing. Personally, I now regard a 50:10:40 split as more appropriate, with the advent of code generators such as those available from Jackson Systems. A phase, represented by a rectangular box, is terminated by a checkpoint (milestone or baseline) shown as an oval box. The circles correspond to review or inspection points which are aimed at verifying that the criteria for a given checkpoint has been realised. This becomes the input to the next phase. A clean compilation is an example of a satisfactory checkpoint, for Inspection I2 in this case.

One of the most important applications of reviews is as an approval mechanism in the management of a project. Review R1 in Figure 1 can improve the future progress of a project by validating the initial technical

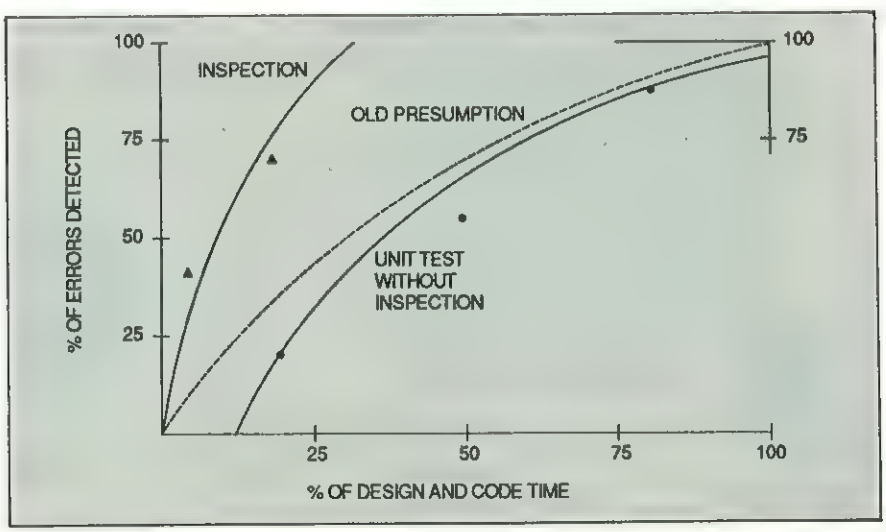
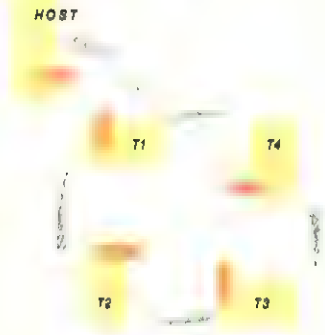
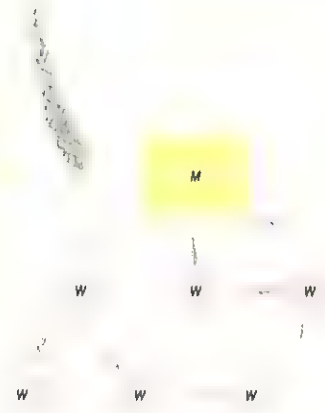


Figure 2 – Comparing Error Detection by Inspection and Unit Tests

THE TRANSPUTER COMPILERS





TRADITIONAL LANGUAGES

Many applications developers world-wide are using the abundant computing power offered by multiple transputer systems. It is sad that many others reject this technology because of the mistaken belief that you can only program transputers efficiently using the occam language. In reality you can write efficient transputer applications entirely in the traditional languages C, Fortran or Pascal using 3L parallel compilers.

EXISTING SKILLS

Transputer applications are built from conventional sequential programs sending data to each other. Such parallel programming needs sequential techniques with which every programmer is familiar. You can use existing programming skills and often even existing code.

FAMILIAR ENVIRONMENTS

Transputers are usually accessed from host computers running standard operating systems. The most common hosts are PCs with MS-DOS but OEM versions exist for others including SUN, Apollo and VAX. 3L compilers and applications developed with them appear as host commands. This allows you to work in familiar environments using host software such as editors and file management utilities.

DEVELOPMENT ASSISTANCE

To assist program development 3L provides Tbug, a powerful source-level debugger. Tbug is an interactive, window-based tool which helps you to observe and control the execution of parallel programs containing many components running concurrently.

FLEXIBLE CONFIGURATION

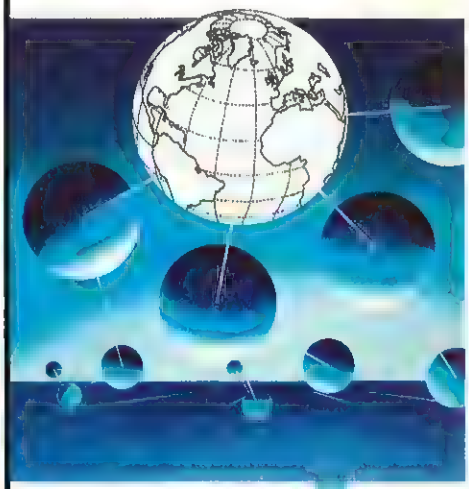
3L's configuration tools let you match applications to existing transputer networks or use processor farm techniques where applications automatically use every available transputer. Processor farms can also be used to run multiple copies of a program in parallel, rather than reinvoking a single instance many times.

WORLD LEADER

The popularity of 3L's transputer development tools, either bearing the 3L name or repackaged under licence, has made them world leaders. 3L software is available from over 70 distributors worldwide.

COMPLETE PACKAGES

3L's compilers and configurers come as complete packages; no extra system software is required to write applications making use of multiple transputer networks. Tbug is available separately. All 3L software is compatible with the majority of transputer add-in boards commercially available.



3L LIMITED
PEEL HOUSE
LADYWELL
LIVINGSTON
WEST LOTHIAN
EH54 6AG
SCOTLAND
TEL 0506 415959
FAX 0506 415944

CIRCLE NO. 764

The Bugbuster.

High-Speed, Source-Level Debug Support
for

C, Pascal and Windows V2.0

Cut development time down to size with ATRON hardware-assisted debug tools.

No more waiting for Watch-Points
debug in high level language with stored trace
while running at full speed.

Debug Device Drivers and Graphics Applications
using ATRON's 'Remote Console' feature.

Track down those elusive memory overwrites
with hardware-assisted debug – allows program breaks
on I/O or memory read/write conditions
plus

Enjoy the security of debug software
protected against overwriting by the program
under development.

Models available for AT and Compaq 386.
Windows upgrade available for existing users.
OS/2 support coming soon.

Distributed and Supported by
QA Training Ltd, Cecily Hill Castle,
Cirencester, Glos. GL7 2EF U.K.
Telex: 437308 Fax: (0285) 650537 Tel: (0285) 655888

QA TRAINING

CIRCLE NO. 765

Figure 3 – A Comparison of Inspection and Unit Test

Types of Error	Inspection	Unit Test
Simple programming mistakes	✔	✔
Logical errors	✔	✔
Development errors	✔	✘
Interface errors	✔	✘
Missing portions of code	✔	✘
Specification errors	✔	✘
Numerical accuracy	✘	✔
Execution errors	✘	✔

specification. It is tremendously worthwhile putting as much effort as possible into this, as many projects have foundered by misinterpreting the original requirement. In the same way, R2 is primarily a validation of the design produced for the system.

Inspections

Here is a technique which has proved to be cost-effective. Some programmers do not see the value of inspections, because they have survived without them in the past or because they think they are suitable only for large projects. This opinion usually changes after a few inspections. Michael E Fagan of IBM (3) claimed that a 23% overall increase in productivity could be realised. Another IBM case showed that the inspection process could detect as much as 82% of errors. This figure has been incorporated into Figure 2. Inspection gives as much as 40% fewer errors than the informal walk-through. Fagan stated that the inspection process should be more than a visual scrutiny and should be a highly-directed and formalised process, maintained by the use of feedback. This involves various forms of follow-up, including telling the programmer the type of errors he is making and how to find/avoid them. Some of the points which a meeting might pursue are:

- Does the code satisfy the requirements of the detailed design specification?
- Inconsistencies in program logic or coding.
- Has provision been made for error conditions specified in the requirements spec, and for run-time errors?
- Does the program include comments of a type and degree that would help future maintenance or modification? This would include cross-references to the detailed design spec.
- How 'estable' is the code under review?

In addition, the team may look for contraventions of sound programming practice, or quality assurance standards. Fagan concluded that code inspection is not only desirable but cost-effective. Clearly the cost of reviews and inspections will be variable. For reviews involving high-level languages, an average rate of 500 LOC (Lines of executable Code) per hour seems appropriate. Low-level code may be three to five times slower. If a one hour preparation time is allowed, then a first review may require one hour/KLOC/person, ie one hour per 1000 lines of code per person on the review. Subsequent reviews of the same code should be about twice that rate. For inspections, the rate may vary between 100 and 150 LOC/hour on a first inspection, but is unlikely to be over 200 LOC. Preparation rate should be about the same as for a review.

The cost of reviews and inspections are roughly comparable because the error detection efficiency of inspections is about double that of reviews. A company might deduce its own rate by plotting records of error count against time/KLOC.

Keeping records of errors allows error detection efficiency to be measured. In future, this could allow the total number of errors to be estimated. For example, if efficiency is known to be 60% and 12 errors are found, then eight remain. Another advantage is that error-prone modules can be identified and appropriate action taken, such as more intense testing and/or inspection of that module.

A comparison of Inspection and Unit Test, in Figure 3, shows that some types of error may not be detected without an inspection process. Figure 2, based on data by Boehm (2) and others, shows another comparison between Inspection and Unit Test. It illustrates the time saved by using Inspection before embarking on actual testing. Here it is assumed that the same length of time has been allocated to testing as was given to design and development. This follows from the 40-20-40 rule previously stated.

In the old days, the curve would terminate at the 100% point (dotted line) as it was

CODE AUDIT CHECKS

Language Audit
- a) conformance to project naming conventions
- b) absence of duplicate names, especially within the same scope.
- c) absence of known error-prone language constructs, eg, data overlays, restrictions on the use of GO TO, 'CODE' inserts.

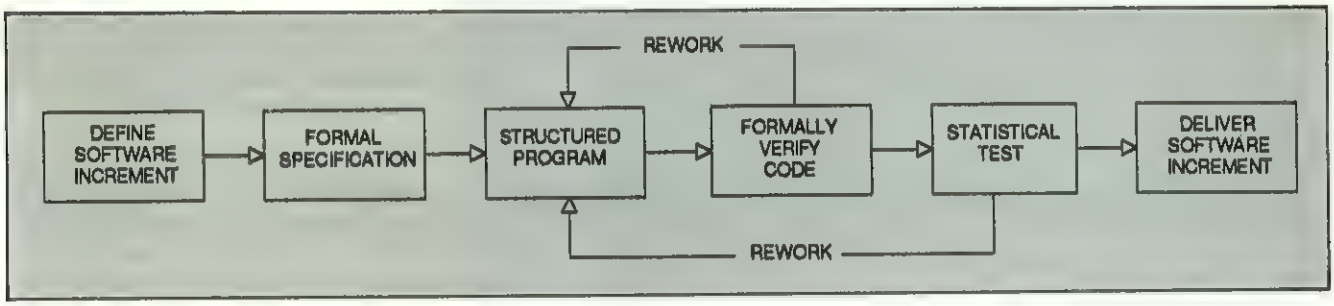
Control Flow Analysis
- a) conformance to recursion conventions.
- b) absence of structurally unreachable code.
- c) absence of structurally non-terminating loops.
- d) absence of multiple entries to loops.

Data Use Analysis
- a) initialisation of data before use.
- b) use of all declared variables.
- c) absence of redundant writes.

Miscellaneous
- a) conformance to methodology specific rules, for example, the MASCOT rules for passing data between activities.
- b) conformance to layout conventions (although a layout editor would be more useful).
- c) commentedness, for example, is the relative ratio of commentary to code above the standard level? Are standard header comments included?

Figure 4 – Checklist for Code Audits

Figure 5 – 'Cleanroom' Software Development Method (IBM)



assumed that testing could detect ALL errors. Presumably, this was because a working program which appeared to work properly was error-free. It is now realised that this is invariably not true.

Some 80-90% of errors may be found for only about 20% of allocated testing time. Unit Test may then uncover a good proportion of remaining errors for another 25% of that time. The use of Inspection before Unit Test has then required only 45% of that time.

Code Audits

Another title for this type of meeting is a Standards and Codes of Practice Audit. Standards may be National/International, and Volume 2 of Reference 4 gives valuable summaries of the most relevant. They may also be contractual, or set by Management. Again, the process will be concerned with errors, but more specifically with the avoidance of methods and constructs which are themselves error-prone. Figure 4 from STARTS (4) shows this and also that consistency and conformance are key features.

Conduct of Meetings

All of these meetings should have an agenda. The first calls for a general functional description, which puts the object to be examined in perspective. A review will then proceed to a walk-through, whereas in an inspection that is replaced by a more specific functional description of the object itself. Inspections and audits cover a wider field and checklists are commonly used to ensure coverage.

A simple agenda for a review meeting could be based on:

- Purpose of object under review.
- Walk-through.
- Summary and conclusions.
- Date and place of next meeting.

It is advantageous to distribute relevant documentation in advance (such as a detailed design spec) for members to study, in preparation for what should be a relatively brief meeting. This will enable members to follow the commentary in the walk-through. Generally, these meetings last for about one hour, seldom more than two, because concentration falls off. It is common practice to limit to two the number of review meetings ON ANY ONE OBJECT. The most appropriate timing for review meetings seems to be when the rate of change in object development appears to slow down.

Those attending walk-throughs should be peers of the person whose work is being considered

The meeting usually begins by the designer, or program author, giving a presentation outlining the purpose of the object under review. This presentation can usually be interrupted on a point of detail, or even to constructively criticise. Fagan suggests that the best size for these meetings is between four and seven members. Their roles might be:

- A chairman or moderator, not personally involved in the project.
- The designer of the object or program.
- The programmer involved in the production of the program.
- The person responsible for testing the program.

If any individual fulfils more than one function, ie a designer who was also the programmer, or the programmer who was also responsible for testing, then someone else should be brought in to fill one function. Sometimes, certain roles are allocated to individual team members. One might look for logical errors, another adherence to quality assurance standards, and another to good programming practice.

Conclusion

It is a fact that these methods work, and that they save money, although their cost-effectiveness may vary from one company to another. It is also factual that they improve productivity, quality and reliability. They tend to be inexpensive to implement, and pave the way for the introduction of other software engineering methods. The methods which have been described could not be automated anyway.

.EXE

John Bruce is a software engineering consultant. He was formerly at National Engineering Laboratory, East Kilbride, where he was Secretary to the Software Engineering Working Party which established software engineering there.

1. Sommerville, I (1989), SOFTWARE ENGINEERING, THIRD EDITION Addison-Wesley, Wokingham.

2. Boehm, B W (1981), SOFTWARE ENGINEERING ECONOMICS, Prentice-Hall Inc, Englewood Cliffs, New Jersey.

3. Fagan, M E (1986), ADVANCES IN SOFTWARE INSPECTIONS, IEEE Trans Software Engineering., SE-12(7), 744-51.

4. STARTS SOFTWARE TOOLS FOR APPLICATION TO REAL TIME SYSTEMS SECOND ED, 1987, Dept Trade & Ind and NCC.

5. Mills, H D, Dyer, M, Linger, R (1987), CLEANROOM SOFTWARE ENGINEERING, IEEE Software, 4(5), 19-25.

PANEL® *Plus* II

Screen Generator and User Interface Library for 'C'

PANEL *Plus* II features: Interactive screen design editor • C code generator • extensive library of user-interface functions • pop-up fields/windows • multiple-line fields • horizontal/vertical scrolling regions • scroll bar/mouse support • help boxes • custom validation • pull-down menus • complete **library source** included • support for popular **graphics libraries** • OS/2 DLL • **no royalties.**

Just released: **Utility Source Option** including source code for the screen layout editor and code generators, which allows **PANEL *Plus* II** to be ported to any suitable development platform. This gives you the freedom to choose and switch your development environment, with total screen and source code compatibilty . Also now supported: Microsoft **FORTRAN** under DOS and OS/2 (the DOS version requires the MS-C run-time library).

Panel *Plus* II can now run on any system with 'C' (but block-mode terminals are not supported). Cost per user: **£295** (including 640-page manual). Please call for quantity pricing, full version list, and free demo disk.

Roundhill Computer Systems Limited

Orchard House, Ogbourne St. George, Marlborough, Wiltshire SN8 1SU
Telephone: **0672 84 535** • Fax: 0672 84 525 • BIX: *join roundhill*

CIRCLE NO. 766

The definitive data protection software for micros and micro-networks

MAIN FEATURES

- All data held on a hard or floppy disk or a network can be kept secure from unauthorised access.
- Security is by default and is failsafe. Once CLAM has been set up the user does not have to take any positive action to secure data.
- Protected files are kept in a permanently encrypted form on the disk. Even if there is a power or hardware failure the data will never be readable by an unauthorised user.
- A log of all activities can be kept.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 01-948 8255.

£148 + VAT Single User.
£580 + VAT per 10 or part of 10 stations on a network.
Site and Corporate Licences available.

CIRCLE NO. 767

Books

This month, reviews of five books.

Preventing Piracy

This book is 4 years old, and American in origin, so some of the figures and most of the law that it mentions will not apply in the UK. However, there are still dozens of useful tips for ensuring that your software (including source code) is only available to the right people, at the right time. Included are sample licence agreements, and advice on how to stop employees and customers from copying your programs, or gaining unduly from your ideas.

Author: Ernest Keet
Publisher: Addison Wesley
Price: £33.95
ISBN:0-201-15047-6
Pages: 150

Programmer's Guide to the OS/2 Presentation Manager

A fairly readable introduction to OS/2 PM programming, for the programmer who already knows how to write in Microsoft C under MS-DOS. A little knowledge of the basics of OS/2, plus the architecture of the 286 is also recommended. What's still scarce is example PM code, and this book contains lots of it. Sybex are obviously competing here with the classic PM programmer's reference from Charles Petzold. The Petzold book is in wide circulation, owing to its free inclusion in the OS/2 SDK. If you don't have a copy, this Sybex alternative is worth considering.

Author: Michael Young
Publisher: Sybex
Price: £24.95
ISBN:0-89588-569-7
Pages: 680

Microsoft QuickBASIC

This pocket reference book lists, with examples, all statements and function calls for QuickBASIC. All versions, up to 4.5, are covered. There is no contents page or index, so this book is only of use if you know the name of the function you want but can't remember the syntax. If you know what you want to do, but don't know how to do it, this won't help.

Author: Kris Jamsa
Publisher: Microsoft Press
Price: £5.95
ISBN:1-55615-204-3
Pages: 135

Out of the Inner Circle

Bill Landreth, also known as 'The Cracker', is just coming to the end of his teenage years. In 1985, when he was 16, he was one of the best known system hackers in the US, and wrote this book as the story of his exploits, and of how the FBI finally caught up with him. Now, this second edition contains most of the original text, plus more up-to-date advice on data security – straight from the horse's mouth, as it were. As well as computer and network security, the subject of viruses is also covered. More of a coffee-table book than a reference manual, and well worth reading.

Author: Bill Landreth
Publisher: Microsoft Press
Price: £8.95
ISBN:0-55615-223-X
Pages: 230

Microsoft C – Secrets, Shortcuts and Solutions

For a change, this C book assumes the reader is using MS C 5.1 rather than QuickC. The difference is really only noticeable in the description of how to actually compile the finished program, but it means that C 5.1 programmers don't have to bother reading about integrated environments that they don't have.

Author: Kris Jamsa
Publisher: Microsoft Press
Price: £21.95
ISBN:0-55615-203-5
Pages: 715

EXE

Looking at dCLIP

Why would anyone want an interpreter for Clipper? There are some good reasons, as Bob Rimmington discovers.

If dCLIP is an interpreter for Clipper, it seems a bit of a contradiction. The whole point of Clipper is that it is compiled. However, that is also a disadvantage. Developing and debugging often require a tedious and repetitive 'compile-link-test-edit' cycle. The basic purpose of dCLIP is to provide an interactive development environment that largely removes this disadvantage.

In essence, dCLIP provides a dBASE-like dot prompt, places your Clipper .OBJ files into runtime libraries and allows routines to be tested one by one. For example, if you write a new function called `MY_FUNC()`, it can be tested from the prompt with:

```
? MY_FUNC()
```

Any relevant variables can be pre-defined and any required parameters passed. It can be used just as if it were a Clipper function such as `DATE()` or `LEN()`. When you are satisfied that it returns the correct values, it can be linked into a .EXE file and used with confidence.

Installation

dCLIP installs itself by unarchiving from two floppy disks into a directory on your hard disk. A minimum of 512K RAM is required, but an 80286, 640K RAM and a RAM disk of at least 384K is recommended.

Using dCLIP

Invoking dCLIP displays a screen with a dBASE III type status bar at the top and the infamous dot prompt at the foot. All Clipper (Summer '87) commands and functions can now be used interactively, though a few limitations are specified. For example, `SUM` and `AVERAGE` are restricted to five fields, `TOTAL ON` will add only one field. The limitations do not apply when the commands are used in a program. See Figure 1 for full details of those commands that are not supported fully.

dCLIP is largely written in Clipper, and this is immediately evident when `HELP`, `BROWSE` and similar commands are entered. It all looks very familiar and may well give you ideas for inclusion in your own applications. The help system comprises a normal .DBF/.DBT combination. It covers all the Clipper commands plus the additional dCLIP ones, and is easy to extend to include any functions in your own library. An editor based on `MEMOEDIT()` is provided, but your own editor can be used if specified in the `DCLIP.SYS` file.

System limits

Maximum of dynamic libraries in use	10
Maximum of mapped DOS commands	20
Maximum of mapped PROCEDURE commands	20
Maximum of mapped BATCH commands	20
Maximum object file size	32

Clipper Commands, differences with interactive use only

?/??	- maximum of 10 expressions separated by commas
APPEND FROM	- record number option for SCOPE not supported
	- FIELDS <field list> replaced with FIELD <field>
AVERAGE	- maximum of 5 expressions
CLEAR ALL	- should not be used as it clears dClip memvars
CLEAR MEMORY	- should not be used as it clears dClip memvars
COPY	- FIELDS <field list> replaced with FIELD <field>
DECLARE	- only one array may be created with each DECLARE
DISPLAY	- maximum of seven expressions in exp. list
INDEX ON ...	- does not create new index if one already exists
JOIN	- maximum of 5 fields in field list
LIST	- maximum of seven expressions in exp. list
PRIVATE	- maximum of 10 memvars for each PRIVATE statement
	- does not create arrays, use DECLARE
PUBLIC	- maximum of 10 memvars for each PUBLIC statement
	- does not create arrays, use DECLARE
RELEASE	- releases only one memvar unless scope options used
REPLACE	- only one field may be replaced for each REPLACE command
RUN/!	- free memory with SET SWAP TO ... in DCLIP.SYS file
SET FORMAT	- used in different way as .FMT file is not compiled
SET INDEX	- maximum of 10 index files may be opened
SET RELATION	- use ADDITIVE to set more than one relation
SORT	- only one field as key, in ascending order only
STORE	- maximum of 10 memvars for each STORE statement
SUM	- maximum of 5 expressions
TOTAL ON ...	- only one field may be totaled for each TOTAL command
UPDATE ON ...	- only one field may be updated for each UPDATE command
USE	- maximum of 10 index files may be opened with INDEX clause

There are no differences in respect of Clipper functions

Figure 1 – dCLIP Specification

The Testing Sequence

The steps required to run or test your own programs comprise:

- Write some procedures or functions (or select from existing ones).
- Compile the file with Clipper to create a normal .OBJ file.
- Create a small ASCII file, listing the .OBJ file(s).
- Run DLIB to create a .DLB file of the same name.
- Enter LIB ‹file name› to load the .DLB runtime library file.
- Call with DO ... or by entering the Function name.

In theory, a complete application could be run in this way, but in practice there seem to be problems if it is of more than modest size. However, there is not really much point to this; the ability to test sections of a program in isolation is far more useful and this works well. Any called functions must be in a loaded .DLB file, of course; any required files must be open and functions from add-in libraries are not supported.

Testing in Practice

For testing, a particularly useful feature is a facility for 'batch' files. These are similar to .BAT files under MS-DOS, but contain a succession of Clipper or dCLIP commands. There is an `IF .... GOLINE` branching facility and this can be used with a simple menu, `GET` and `READ`. While developing and testing an application over a number of days, a batch file could be used to open files and initialise variables. Just like a .BAT file, it handles parameters passed as %1, %2. It can load a .DLB file and then call functions in it.

A good illustration of dCLIP in action arose during my testing of a complex calculating function. The aim was to forecast processing time on a range of machines for varying job specifications and quantities. The main function called several others, received data as parameters and read fixed data from a .DBF file that had a record for each machine. The first stage was to call both the subordinate and main functions to check that they worked and returned sensible figures. Next, a batch file was written to call the main function repeatedly, but with different system variable values and different parameters. To move between records or change data in the file, `BROWSE` was used. A further function used the returned values to select the most efficient machine for each task. All the output could be directed to a file and passed to the client, to verify that the program was producing the right answers and making the correct decisions. To do this without dCLIP would have required temporary extra routines for data input and reporting plus, of course, the 'compile-link ... cycle as each error was found. It must be said that dCLIP proved the ideal tool for this task.

Figure 2 – dCLIP Additional Functions

```
ADD_REC(<expN1>,<expL1>)
BUILD_COND(<expC1>)
CD(<expC1>)              ** Part of OVERLAY **
CURPATH([<expC1>])  ** Part of OVERLAY **
CHR_SEL(<expN1>)
COLOR(<expC1>)
COLOR_SEL(<expn1>)
DBCHOICE( 26 parameters )
DBFILE(<expC1>,<expC2>,<expL1>,<expL2>,<expN1>)
DBINDEX(<expC1>,<expC2>,<expC3>,<expL1>,<expL2>,<expL3>)
DBPACK(<expC1>,<expC2>,<expC3>,<expL1>,<expL2>)
DBSELECT(<expC1>)
DBSTRU(<expN1>,<expN2>)
EXPLODE(<expN1>,<expN2>,<expN3>,<expN4>,<expC1>,<expC2>,<expL1>)
GETVAR(<expC1>,<expN1>,<expC2>)
IMPLODE(<expN1>,<expN2>,<expN3>,<expN4>,<expC1>,<expC2>,
        <expL1>,<expC2>)
INKEY_PROC(<expC1>,<expN1>)
KEYDISP()
KEYEDIT()
KEYLOAD()
LIBLOAD(<expC1>)
LIBLOADRUN(<expC1>,<expC2>,<expL1>)
LIBUNLOAD([<expC1>/'ALL'])
MEMOPRINT(<expC1>,<expN1>,<expN2>,<expL1>,<expC2>)
MEMOVIEW( 15 PARAMETERS )
NET_USE(<expC1>,<expL1>,<expN1>)
OBJLOAD(<expC1>)
OBJLOADRUN(<expC1>,<expC2>,<expL1>)
OBJRELEASE([<expC1>/'ALL'])
OVERLAY(<expC1>,<expN1>,<expC2>,<expC3>,<expC4>,<expL1>)
PR_TEST()
PR_CONT()
REC_LOCK(<expN1>,<expL1>)
SETINIT()
SETREAD([<expC1>])
SETRESTORE(<file>)
SETSAVE(<file>)
SETWRITE(<expC1>,<exp>)
SET_PRINT(<expC1>,<expL1>,<expL2>,<expL3>,<expL4>)
TESTFILE(<expC1>,<expL1>,<expL2>)
```

Interactive Tools

All the dBASE commands such as `EDIT`, `DISPLAY`, `MODIFY STRUCTURE` are supported, including `ASSIST`. Some, but not all, are quite similar in appearance to the dBASE originals. `ASSIST` is described as a 'Menu-Driven Database Manager' and it looks like just that. There is a quite separate `QUERY` command which enables a complex filter condition to be built through menus. Further tools are supplied to design report and label forms. They are all adequate for development purposes but, for anything more than straightforward queries and reports, you would do better with R&R Report Writer (as mentioned in the December 1989 issue of .EXE). In contrast, the `BROWSE` command is far superior to the dBASE equivalent. It would appear to have been written with the Clipper `DBEDIT( )` function and is a good demonstration of the scope it offers. Pop-up menus, search facilities and edit options are all included.

Three new `KEY` commands allow definitions for 40 function key combinations to be placed in a DCKEY.DBF file. `KEY LOAD` will activate it, `KEY EDIT` enables definitions to be entered or amended and `KEY DISPLAY` shows a summary table. Another handy extra is `CHR` which pops up tables of all 256 ASCII characters and allows selection with the cursor. A `PRINTER MANAGER` maintains a .DBF file of

code sequences for a range of printers and enables new records for unlisted printers to be added.

Additional Functions

There are 40 functions, and these are listed in Figure 2. Some, such as KEY EDIT( ) and CHR_SEL( ), are equivalent to the corresponding commands. Others, such as DBCHOICE( ), are enhanced alternatives to corresponding Clipper functions. COLOR_SEL( ) provides a useful panel where colours can be selected with the cursor. All these functions are supplied in both a runtime .DLB file and in a conventional .LIB file for linking into your .EXE file, so you could include a routine that enabled a user to select his own colour preferences. It is a pity it cannot be used on dCLIP itself to change combinations such as yellow on magenta.

BUILD_COND( ) simplifies design of a user query tool as it incorporates pop-up pick lists for fields and conditions. SET_PRINT( ) takes parameters appropriate to print styles, refers to the .DBF file defined with the PRINTER MANAGER and sends the required code to the selected printer. OVERLAY( ) allows memory to be released from a Clipper application 'on the fly' and saved to a temporary file.

Other Features

As several runtime .DLB files can be loaded simultaneously, it is possible to lose track of just what is in use. VIEW LIB displays a table showing the current position, with data for each .OBJ file that includes its size. Further .NAM and .LST files corresponding to each .DLB file provide size, date and time listings of OBJects and a table of the included procedures and functions.

What is described as an Interactive Debugger is provided. In fact pressing Alt-Left Shift suspends execution, rather like dBASE III+ does when it encounters an error. It is then possible to use most of the normal interactive commands to look at variables, inspect the database and so on. With caution it is also possible to make changes, such as revising values or creating an index, before resuming operation by pressing ENTER. It is not possible to single step through a program, but SET TALK ON and SET TRACE ON commands are available.

A HISTORY buffer of 200 command lines is maintained, which avoids the tedium of retyping repetitive commands. Recalled lines may be edited before re-use. A DUMP command displays a file in both hex and ASCII form, BATCH runs a batch file. SEARCH will look for a text match through all fields in one or more databases.

The Down Side

dCLIP is not without its shortcomings. Although the manual contains most of the guidance needed, the general presentation is poor. The text is cramped (10 lines and 14 characters to each inch) and yet it is set on sheets with excessively wide margins all round. There is no tutorial and no index.

All output could be directed to a file and passed to the client for verification that the program was producing the right answers and making the correct decisions

Getting started involves a deal of back and forth searching through the first few chapters.

Memory, or rather the lack of it, can also be a problem. dCLIP itself requires 350K plus a minimum of perhaps 24K workspace. That leaves barely 200K for your own Clipper programs, libraries and workspace. A maximum of 32K is advised for any program called with a DO though no limit is specified for the .DLB files. It is said that a complete application can be run from within dCLIP and, maybe with perseverance, it can. Mine created .DLB files totalling 150K and failed repeatedly. To be fair though, this was attempted only for the purposes of the review and I would not visualise a need to do this in normal use. Work on selected fragments presented only one problem; if functions call others, such as ones you hold in a STANDARD.OBJ file, you must first identify, extract and attach these to the test routines.

For editing, compiling and similar operations, adequate memory is made available by the swap-to-disk facility. This worked well with a RAM disk with one proviso. If the space available is less than the swap space specified, the commands just do not work and no message is displayed. Even less helpful, if your DCLIP.SYS file does not end in a Ctrl-Z, your PC will just hang when dCLIP is started.

Error Reporting

The manual makes no mention of the above errors and very little about the error messages you will get. These must be among the most unhelpful seen with any software. For example:

```
msg=ZEROLEN prg=[obj2lib]:
        P1='DEC2REAL.obj'
msg=PINIT prg=[c_call]: ERROR
msg=MALLOC prg=[kern,s_obj2lib]:
        ERROR
```

It would seem that dCLIP traps the normal Clipper error system to substitute these gems. One test involved an .OBJ file that included my own error trapping functions. A .DLB file was created quite normally, but not one of the functions in it was recognised. Removing the error routines solved the problem.

Conclusions

Is dCLIP a worthwhile tool for a Clipper programmer? Despite the rough edges, yes, I think it is. Some prior experience with dBASE and Clipper will enable you to stumble through the initial familiarisation steps without too much difficulty. Similarly you will already know how to interpret, for example, the implication in the manual that you can both SET CLIPPER=R10 and have 2048 memory variables. To test your application as a whole, you will probably still have to compile and link as before, but by then you should have identified most of the bugs. Calculations, screen layouts, pick-lists, reports and virtually every other component can be pre-tested before they are finally all brought together. For all but the most perfect programmers it should save a great deal of both time and frustration. Indeed it could be argued that Clipper now offers a better development environment than an interpreter alternative.

Bob Rimmington is an independent dBASE and Clipper consultant. He can be contacted through Stanford Systems on 0444 236352. dCLIP may be obtained from Software Paradise, Avenue House, King Edward Avenue, Caerphilly, Mid Glamorgan, CF8 1HE. Tel: 0222-887521, Fax: 0222-862209. The version tested was 1.15 and is priced at £179.

MKS Toolkit | **XENIX** | **MICROPORT SYS V/286 £395.00 Complete Systems V/386 £495.00** | | **MINIX**

XENIX & UNIX

SCO XENIX	286	386
SCO Xenix Op Sys	£357.00	£422.00
SCO Xenix Dev Sys	£357.00	£455.00
SCO Xenix Text Proc	£126.00	
SCO Xenix Comp Sys	£775.00	£905.00
SCO Unix (2 User) Op Sys		£341.00
SCO Unix Devel Sys		£568.00
SCO CGI Graph Run/Sys	£191.00	
SCO Multiview	£256.00	£321.00
SCO VP/ix		£321.00
SCO Foxbase+	£516.00	£647.00
SCO Foxbase+ Run/Sys	£191.00	£191.00
SCO Masterplan	£516.00	
M/Soft Word	£386.00	
Multiplan	£321.00	
SCO Eroff Documentor	£451.00	£516.00
SCO Lyrix v6.0	£386.00	£386.00
SCO Professional	£386.00	£386.00
SCO Image Builder	£386.00	
M/soft BASIC compiler	£451.00	
M/soft BASIC interp.		£256.00
M/soft FORTRAN comp.	£451.00	
M/soft PASCAL comp.	£451.00	
SCO XENIX-NET Server	£386.00	£451.00
SCO XENIX-NET 3 Server	£841.00	£1036.00
SCO TCP/IP (Req streams) Run/Sys	£227.00	
SCO TCP/IP (Req streams) Dev/Sys	£146.25	

Microport UNIX	286	386
Complete System	£395.00	£495.00
Comp. Sys. Unlimited	£538.00	£718.00
Runtime System (2 user)	£159.00	£189.00
Run. Sys. Unlimited	£299.00	£358.00
Unlimited User Upgrade	£159.00	£189.00
Software Development	£189.00	£329.00
Text Preperation	£129.00	£129.00
Network Services Extension (Includes Streams & RFS)		£119.00
DOS Merge Limited	£165.00	£249.00
DOS Merge Unlimited (Requires relevant Runtime Sys)	£165.00	£319.00
RM/Cobol Devel	£750.00	
RM/Cobol Runtime	£169.00	
RM/Fortran Devel	£450 00	
M/Soft Basic Compiler	£357.00	£417.00
WordPerfect		£597.00
Twin/UX - Spreadsheet		£417.00
Informix		
4GL Devel	£8970.00	£1497.00
4GL Runtimes	£450.00	£750.00
SQL Devel	£717.00	£837.00
SQL Runtime	£357.00	£417.00
ESQL/C Devel	£537.00	£657.00
ESQL/C Runtime	£270 00	£330.00
C-ISAM	£199.00	£229.00
Unify		
Accell IDS Devel	£1797.00	£2700.00
Accell IDS Runtime	£900.00	£1350.00
Unify DBMS Devel	£597.00	£1197.00
Unify DBMS Runtime	£297.00	£600.00
TCP/IP Board+Software Driver	£894.00	£894.00
TCP/IP Software Driver Only	£357.00	£357.00
Complete System V Manual Set	£119.00	£149.00
Individual Manuals Each.	£60.00	£60.00
User's Guide		£60 00
Advanced Programming Guide		£60.00
Microport System V Made Easy		£25.00

Sys-IX Programs shown below run on both XENIX, UNIX and DOS Operating Systems. You can either design your own custom built application or have our highly trained in house personell do it for you...

Most Major Hardware Brands Sold. £prices.00 too low to advertise. Please call for

The GREAT CD-ROM Bundle

	RRP	Ours
Hitachi 1503S Ext. (Card&Cable)	£630	£550
Hitachi 1503S Ext. MCA	£700	£620
Hitachi 3500 Int. (Card&Cable)	£600	£525
Philips CM121 PC/XT/AT (C&C)	£675	£576
Philips CM201 Internal-as above	£625	£525

Microsoft Bookshelf £279.00 *You get instant access to: The World Almanac, Chicago Manual of Style, Bartlett's Familiar Quotations, Roget's II: Electronic Thesaurus, American Heritage Dictionary, Business Information Sources, Houghton Mifflin Spell Checker and Usage Alert and more!*

Microsoft Programmers Library £345.00 *A Complete library (over 20,000 pages) of the latest releases of Microsoft's Technical Reference Manuals covering S/2, Windows MSDOS, C, MASM, etc with 8 megabytes of source code.* ***Also includes FREE CD ROM Networking Software . PC Professionals need this NOW!***

THE BUNDLE For **£895.00**, you get Microsoft Programmers Library, a coupon to buy both Micrsoft Stat Pack & Small Business Consultant for only £50.00 each, and a **FREE** copy of CD-Play Demo.

OPERATING SYSTEMS

UNIX Sys V/286 Complete Sys Microport	£395.00
UNIX Sys V/386 Comp. Sys M.port	£495.00
XENIX 286 Complete System by SCO	£775.00
XENIX 386 Complete System by SCO	£905.00
386/IX TCP/IP by Interactive System	£379.00
386/IX (2 User) Complete Interactive System	£995.00
ESIX/V 386 w/Dev.(3 User) by Everex	£769.00
QNX Op Sys by Quantum Software	£450.00
Concurrent DOS 386 (3 User) by Digital Research	£315.00
Concurrent DOS 286 (3 User) by Digital Research	£235.00

SOFTWARE

Sys-IX, Single User DOS	£395.00
Sys-IX, Multi User /Novell/OS2	£595.00
Sys-IX, Single User XENIX	£495.00
Sys-IX, Multi User XENIX	£795.00
SOURCE available	call .
TERMINAL CONTROL with source	£895.00
SYS-IX Editor DOS version	£49.95
SYS-IX Editor XENIX version	£49 95
SYS-IX Editor Custom simulate Word Star	£75.00
SYS-IX Editor Custom simulate VI editor	£75.00
SYS-IX Editor Available in 5 languages: English, German, French, Spanish, Latin	£75.00
SYS-IX Spell 30,000 Words	£25.00

UTILITIES & TOOLKITS

ARTIFICIAL INTELLIGENCE File Based	£49.95
AUTO ROUTE Fastest, shortest, scenic journies	£150.00
Brief Editor	£149.95
BTree Isam by SoftFocus - Multi User	£145.00
CB+Trees	£149.95
GURU Gen User Reporting Util	£49 95
KEYPILOT - Demo trainer utility	£129.95
MENU - IX User Defined Interface	£49.95
ME Programmers Editor with Macros	£139.95
MINIX Op/Sys UNIX like Op Sys	£149.95
MKS Tool Kit Unix Environment	£149.95
OOPS Collection of C++ classes	£49.95
PC EXPERT Expert System	£249.95
POLYGLOT TSR PACKAGE V.Good utils	£75.00
R.Doc X WStar, DWrite 4, Word. Switch & Ref Files	£149.95
R.Tree Report Generator	£235.00
D.Tree Screen Handler	£375.00
C.Tree File Handler	£295.00
SLATE - Painter Routine in C	call .
SPELL CHECKER	£49.95
TAWK Enhanced like Xenix awk	£49.95
TSR SOFTWARE KEYBOARD LOCK	£10.00
Turbo G Graphics Library	£135.00
TURBO TEX	£135.00
VITAMIN C Program Generator	£199 95
AT/XT BIOS Kit Your own BIOS, in/out comp set.	£160.00

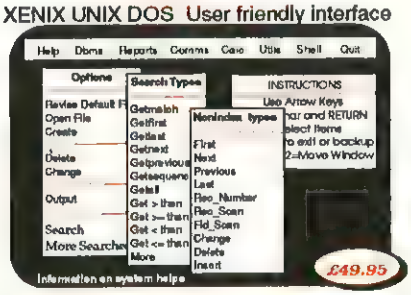
POPULAR SOFTWARE

ABC Bus System	£145.00	Quick Basic v4.5	£65.00
Aldus PgMaker 3.0	£540.00	Quick C v2.0	£65.00
Ami Professional	£345.00	Quick Pascal	£65.00
C Compiler 5.1	£239.00	R-Base Dos v2.1	£377.00
Chart v3.0	£193.00	R-Base Tools	£79.00
Clipper Summer 87	£325.00	Sage Accountant	£169.00
DBase III+ v1.1	£325.00	Sage Accountant+	£279.00
DBase IV	£375.00	Sage Bookeeper	£75.00
DBase IV Devel	£599.00	Sage Fin Controller	£349.00
Draw Plus v1.0	£220.00	Sage Job Costing	£150.00
D/Tool Framework 2	£79.00	Sage Retrieve III	£119.00
Framework III v1.1	£368.00	Smart II Full Sys	£429.00
Graph Plus v1.3	£290.00	Smartcom III	£119.00
Har/Graphics v2.1	£412.00	Smarterm 220	£129.00
Logistix v1.31	£119.00	Turbo C v2.0	£72.00
Lotus 123	£269.00	Ventura Prof v2	£298.00
Lotus Symphony v2	£359.00	Ventura Pub v2	£620.00
Lotus Freelance	£269.00	Windows 286	£65.00
Lotus Agenda	£221.00	Windows 386	£119.00
Lotus Magellan	£92.00	WordPerfect v5.0	£299.00
Manuscript v2.1	£289.00	Wordstar Prof 5.01	£249.00
Microsoft Word v5	£299.00		
Micrographix Desig	£387.00		
Paradox 386	£352.00		
Paradox v2.01	£294.00		
Paradox v3.0	£407.00		

We can supply many other software packages not shown. Please call for our £prices.00, you will be *pleasantly surprised...!*

MENU - IX

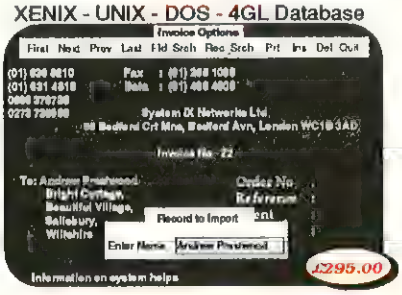
XENIX UNIX DOS User friendly interface



Powerful, friendly, colourful, complete menu system that you can customise to exacting requirements

SYS - IX

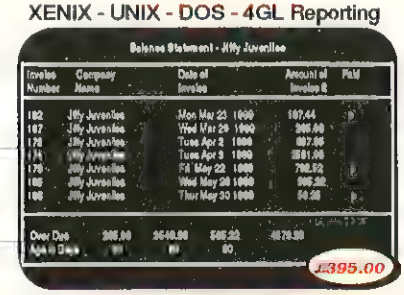
XENIX - UNIX - DOS - 4GL Database



A powerful & complete applications generator for forms & database management, inlcudes Menuix.

SYSTEM - IX

XENIX - UNIX - DOS - 4GL Reporting



Powerful toolset for reporting database file systems under Menuix & Sysix

All prices exclude VAT

VISA ACCESS

System IX Networks Ltd
11 Bedford Court Mansions
Bedford Avenue
London WC1B 3AD
Tel Sales: 01 - 636 8210 (6 lines), Customer Services: 01 - 631 4818, Fax: 01 - 150 1088

Information Pack
Programming Utilities

Price List
Sales Enquiries
Demonstrations

CIRCLE NO. 769

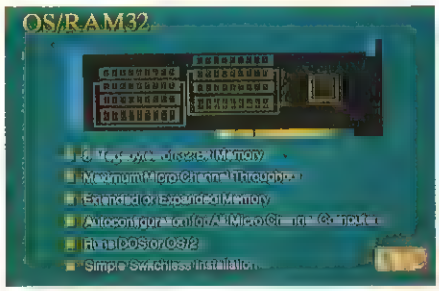
Introducing OS/RAM32

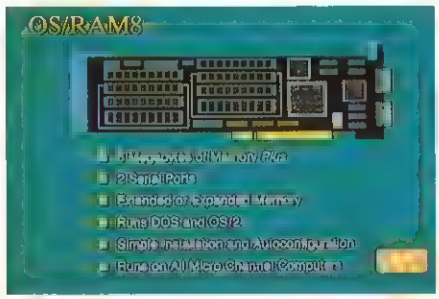
The Only High Speed 32-bit Memory Card That Works in ALL MCA Computers. Automatically Configuring its Bus Width for 16-bit and 32-bit MCA Slots!

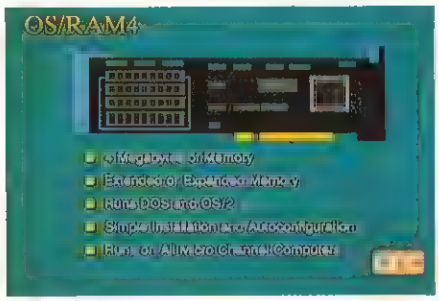
- Single high speed memory card for all your needs.
- ✓ Advanced SINGLE chip design.
- Clock rate of 70MHZ for fastest MCA computers.
- Single card supporting up to 8MB of fast memory.
- Extended and Expanded memory (LIM 4.0).
- Expanded memory many times faster than Intel.
- ✓ Compatible with all operating systems.
- Auto-configuration at power up (like IBM BIOS).
- ✓ Compatible with all MCA features.
- ✓ Easy Installation.
- ✓ IBM approved ID number.
- Preserves your investment now and in the future.
- ✓ Two year Warranty.
- Full range of memory cards to satisfy all budgets.
- ✓ Unique feature.

  Use any OS/RAM product in any MCA computer.

  (ieApricot Qi,Olivette P series,RM PS/2)







OS/RAM32 £325

OS/RAM8 £292
(plus two serial ports)

OS/RAM4 £229

OS/RAM 4 - OS/RAM8 - OS/RAM32 prices do not include memory.

PLUS

RamStar. Full range of 100% compatible Compaq memory modules and cards. Only highly reliable Siemens memories are used. 1MB,2MB,4MB for 386/33 386/20 386/20e 386/25 386/s 286e SLT

ORDER YOURS TODAY TEL: 01-938-2151

TEL: 01-937-4643 FAX: 01-937-2579

PLEASE ADD VAT TO ALL PRICES.

Sole UK Distributor

ADVANCE MICROPROCESSOR DEVELOPMENT SYSTEMS Ltd

7-11 Kensington High Street
London W8 5NP,UK

PS/2 is trade marks of IBM. Olivette,Apricot, Siemens and Compaq are trade marks of their respective companies.

Please send me more information on

OS/RAM32 ☐

OS/RAM4 and OS/RAM8 ☐

Compaq memory modules/cards ☐

Name :.....

Company:.....

Address :.....

.....

.....

.....

Telephone no:.....

CIRCLE NO. 770

A Taste of Icon

Icon looks rather like yet another ALGOL variant on the page, but beneath this bland exterior lurk some powerful mechanisms. Paul Smith explains.

Icon is a general-purpose programming language that is especially suited to text processing and non-numerical applications, ranging from symbolic algebra through to natural language processing. It is designed to be easy to use: suitable for problems that require a 'quick and dirty' solution and powerful enough to handle complex applications.

The Icon language is currently in its seventh major version, and is still evolving under the control of Ralph E Griswold at The University of Arizona. Icon originates from the SNOBOL family of languages, which was designed and implemented at Bell Telephone Laboratories during the 1960s and 1970s by a team that included Griswold. SNOBOL, like Icon, is a high-level programming tool for manipulating strings of characters that emphasised ease of programming. The best-known version of SNOBOL is SNOBOL4 (1971), which to this day is in use on a wide variety of computers – in spite of an early stated view of a certain Professor of Computer Science that it didn't stand a snobol's chance in hell.

Griswold continued to work on the SNOBOL language after moving to the University of Arizona, completing Version 5 of the language in 1975. However, the next year he realised that the language elements and control structures for string processing could be fused into a single coherent programming language. (In SNOBOL, pattern matching and string handling facilities are implemented as a sub-language.) Icon was born.

Icon Programs

To quote Griswold: 'Icon programs consist of declarations and expressions. The declarations define information that must be known before program execution begins, the expressions constitute the executable part of the program.'

Icon programs contain one or more procedures, together with definitions for any other global identifiers that are required. One of these procedures, in which program execution begins, must be called `main()`. The Icon version of the triangle program appears in Figure 1. *(Please see the July 1989 issue of .EXE Magazine for details of the triangle problem).*

Icon provides an extremely rich set of operators, including no less than three different equality operators

At first sight Icon appears similar to languages like C, Pascal, and ALGOL 68. However, Icon has many special characteristics that these others do not share. At the heart of the Icon language is the concept that expressions may succeed or fail. For instance, in Icon, the comparison

```
a > b
```

will succeed if the value of `a` is less than the value of `b`, and fail otherwise. Likewise, the operation

```
find("two", "in northern areas")
```

will fail whereas

```
find("off", "the kick-off starts")
```

will succeed.

Icon has no concept of Boolean or logical values. Instead, control flow is influenced by the success or failure of expressions. Furthermore, an expression that succeeds will return a value, but an expression that fails will return nothing. If an expression that fails is required to evaluate another expression, the larger expression will not be evaluated and it too will fail. Thus

```
a := 220 < r
```

will succeed if `r` is greater than 220, and the value (`r`) of the expression will be assigned to `a`; if `r` is less than or equal to 220, or cannot be evaluated as a numeric value, the expression will fail and nothing will be assigned to `a`.

Icon offers the usual variety of control structures, such as `if expr1 then expr2 else expr3`, `case expr of case- clause, ...`. There are also iterative mechanisms: `every expr1 do expr2`, `while expr1 do expr2`, and `until expr1 do expr2`.

Data

Icon offers 11 built-in data types: null, integer, real, string, cset, set, list, table, co-expression, procedure and file. Icon variables can have values of any type. Structures may contain elements of different types, without restriction. When necessary Icon will execute type conversions – if a conversion cannot be performed, the expression that requires it will fail.

Csets and sets are similar to each other. Csets are unordered collections of charac-

Figure 1 – The Triangle program

```
# program: Triangle.Icn
# Third Side program in Icon for .EXE Magazine

procedure main()
   triangle := list(3,0)
   every i := 1 to 3 do triangle[i] := getSideLength(i)
   writes("Triangle entered is: ")
   every writes("   ", triangle[1 to 3])
   write()
   if isatriangle(triangle)
      then {
         if isequilateral(triangle)
            then write("This is an equilateral triangle")
         else if isisosceles(triangle)
            then write("This is an isosceles triangle")
         else write("This is a scalene triangle")
      }
   else
      write("This is not a triangle")
   write()
end


procedure getSideLength(i)
   local s
   repeat {
      writes("Enter side length ", i,": ")
      s := read()
      if numeric(s) then break
      else write(s, " - you must enter a number: try again")
   }
   return s
end

procedure isatriangle(t)
   if (t[1] <= (t[2] + t[3])) &
      (t[2] <= (t[3] + t[1])) &
      (t[3] <= (t[1] + t[2])) then return
   else fail
end

procedure isequilateral(t)
   if t[1] = t[2] = t[3] then return
   else fail
end

procedure isisosceles(t)
   if (t[1] = t[2]) |
      (t[2] = t[3]) |
      (t[3] = t[1]) then return
   else fail
end
```

ters, whereas sets are unordered collections of any type or types of data. The usual set operations (inclusion, exclusion, intersection, union, and so on) may be performed on csets and sets. Strings are ordered collections of characters: they are not arrays, although individual elements can be accessed directly. There is no separate data type for single characters.

Lists are ordered collections of data, elements of which may be directly accessed. They can be used like vectors, stacks and queues. Lists can contain references to values of any data type, including themselves.

Tables offer an associative access mechanism: each entry is associated with a subscript. Both table entries **and** subscripts can be values of any data type. When a table is created a default value for new entries is specified. So

```
t := table(0)
```

defines a table in which new entries will be initialised with the value zero. Individual entries may then be created and assigned thus:

```
c["begin"] := 7
t["if"] := 15
```

Procedures are first class data values. They can be assigned to variables and passed to, and returned from, functions. Icon does not go as far as LISP in this respect, though: there is no mechanism for constructing new procedures during program execution. Icon's versatile type conversion facilities allow some strange goings-on. Given the pre-defined procedure `write()`, one can do this:

```
gargle := write
```

and then use `gargle()` interchangeably with `write()`. Thus

```
s := "throwback"
gargle(s)
```

will write the string value in variable `s` to standard output. Icon can convert string values into identifiers, so

```
"+"(i, 4)
"write"(i)
```

and

```
i +:= 4
write(i)
```

are equivalent.

Icon provides an extremely rich set of operators, ranging from arithmetic, through set manipulations, to a multitude of conditional assignments. To give you an idea of the variety, you should know that there are no less than three different equality operators: = compares two numeric values, == compares two string values and === compares both value and data types at once.

Strange expressions

In most computer languages, a given expression should generally return a fixed result, how ever many times you evaluate it. With the notable exception of random number generators, expressions that change value are frowned upon. Not so in Icon. Expressions may produce more than one result (generators), or they may produce no result at all.

Each time they are invoked, generators produce the next result in a sequence, until there are no more results and they fail. For example

```
find("a", "What a lot")
```

is capable of producing two results. This loop

```
every write(find("a", "What a lot"))
```

prints the results '3' and '6'. Other generators include expressions like

```
8 to 24 by 2
```

which generates the integers from 8 to 24 in steps of 2.

Generators can be used in two distinct ways: in iterative expressions, like the `every` loop above, and in goal- directed evaluation. The latter refers to expressions where, although they might fail at first, repeated calls to the generator in the expression can lead to success. Procedures may act as generators. A reserved word `suspend` is used in a procedure to return a value and suspend processing until its next invocation, retaining the values of all local variables.

Icon allows the programmer to declare and use 'co-expressions'. These can be thought as 'generators assigned to variables'. Co-expressions reproduce a previously captured sequence of results. Here is a co-expression to generate a sequence of numbers:

Figure 2 – Word count program illustrating string scanning

```
procedure tabwords()
   static wchar
   local words, line
   initial wchar := &lcase ++ &ucase ++ '\'-'
   words := table(0)
   while line := read() do
      line ? while tab(upto(wchar)) do
         words[tab(many(wchar))] +:= 1
   return words
end
```

`nseq := create (5 to 9999 by 3)`
It is activated using the @ operator. This code fragment
`every write(@nseq)`
prints the sequence of numbers to standard output. The^ operator refreshes (ie re-initialises) co-expressions. For instance,

mseq :=^ nseq
places a copy of nseq into variable mseq. Activating mseq will produce the same sequence of numbers, from the beginning.

An important feature of Icon is the string scanning facility. This is enormously powerful, and it lends itself to many kinds of text parsing and language processing applications. An expression of the form
`str ? expr`
(where `expr` may be a compound expression, delimited by { and }) defines a string (`str`) on which the expression operates. Within the expression will be functions that analyse the string. These functions can also produce sub-strings that may be assigned to variables or processed in other ways. For instance, the function in Figure 2 creates a table of words and their frequencies within the input file.

Running Icon

Icon requires a large amount of run-time support, and is implemented as a paired language translator and run-time system. The translator reads source code files, and translates them into an intermediate code format which is then executed under the control of the run-time system. Implementations are available for a variety of hardware and operating system platforms, including Unix systems, the Macintosh, and MS-DOS computers. I used the MS-DOS flavour of Icon, version 7.5, during the preparation of this article.

EXE

Paul G Smith is a freelance software development consultant, specialising in graphics, communications, and the application of object-oriented programming techniques. He can be contacted on CIX as "pgsmith", and on AppleLink as "UK0310".

He recommends the following books for further reading:

The Icon Programming Language *(Griswold & Griswold, pub Prentice Hall, ISBN 0-13-449777-5). This discusses V5.0 of Icon.*

The Implementation of the Icon Programming Language *(Griswold & Griswold, pub Princeton University Press, ISBN 0-691-08431-9). This discusses V6.0 of Icon.*

The University of Arizona's public domain MS-DOS implementation of Icon, together with further documentation of the language, may be downloaded from the CIX topic 'icon/msdos'.

Recursion: see recursion

Recursion may be a good source of jokes, but most compiler manuals warn you that its use is bad practice. John Healy, however, thinks otherwise.

Too often, I have heard people who write programs condemn recursion techniques as the playthings of theoretical computer scientists, using words like 'inefficient', 'obscure' and 'difficult' to drive the point home. It is not difficult, however, to show that recursion is one of the most powerful programming tools we have. Not only can it improve the efficiency of programs (as determined by their execution speed) but it can also improve their clarity.

Well written recursive routines are fast, elegant and easy to understand. In simple programming terms, recursion is where a routine P calls itself as many times as is necessary until some goal is reached, after which, the recursion is 'popped off' as the routine returns occur.

Many functions may be expressed nicely in a recursive manner. Indeed, it may be argued that recursive functions are 'nicer' than recursive callable subroutines because their calls are not explicit (ie they occur as part of expressions).

Consider as an example the factorial function fac(x), or (x!). Here are two ways of expressing the function (this pseudocode assumes all variables are non-negative integers):

The recursive code contains an embedded semantic. The code describes what a factorial is, rather than how to compute one

Program 1.

```
function fac (x)
 fac := 1;
  while i⩽ x
   fac := fac * i;
   i := i + 1
  wend
end function
```

Program 2.

```
function fac (x)
 if x = 0 then
  fac := 1
 elseif x › 0 then
  fac := fac * fac (x-1)
 endif
end function
```

I believe that method 2 is a better way of implementing the factorial function. In method 1 the function is expressed in a very 'computational' manner. The piece of code describes an algorithm which must be examined very carefully to determine its meaning. Method 2, however, contains an 'embedded semantic'. The meaning of the piece of code is much more intuitively obvious. It describes what a factorial is, rather than how to compute one. It contains information about computational intent rather than just computation method.

These embedded semantics are, I believe, the best reason (under certain circumstances) for choosing recursive over iterative function implementations. Well chosen recursive functions carry more information from the specification to the implementation than non-recursive algorithms.

One area where the power of recursion may be seen quite clearly in a realistic situation is that of text substitution. A problem is defined as follows:

Write a program to read an ASCII text file and replace all occurrences of string x with string y, and write all the new text to a new file.

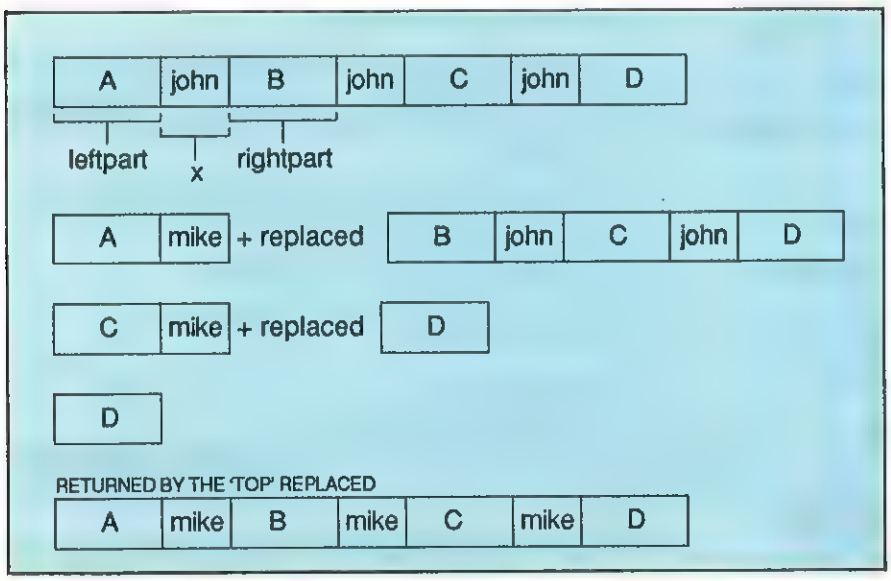
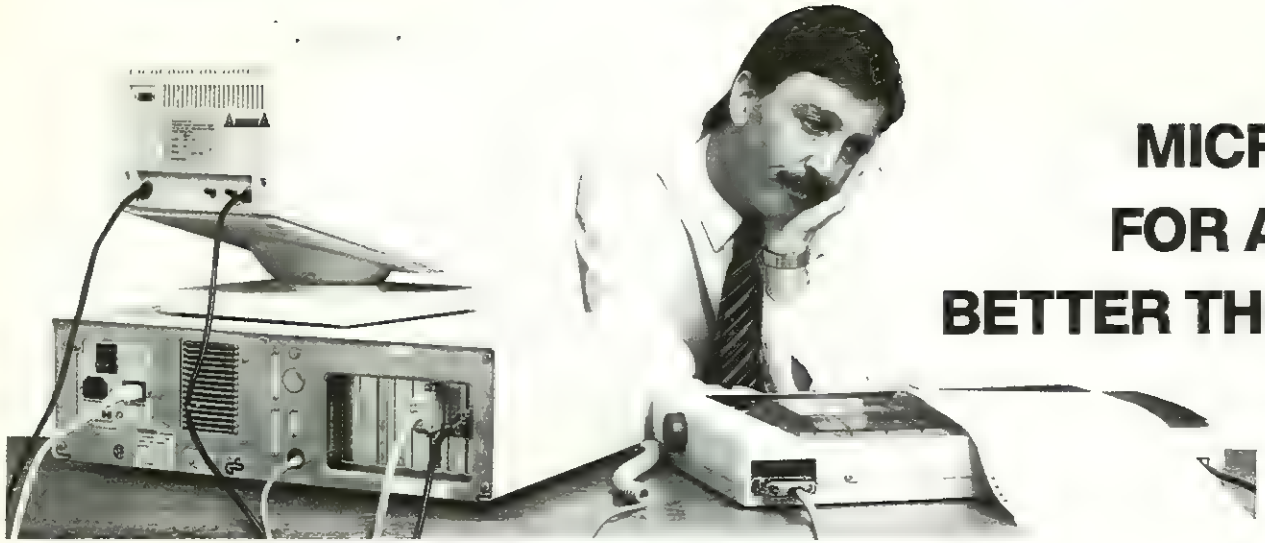


Figure 1 – How the replaced() function works



X-BUFFER FROM MICRO CONTROL SYSTEMS FOR ALL THOSE WHO'VE GOT BETTER THINGS TO DO WITH THEIR PC THAN WAIT FOR THEIR PRINTER

Imagine you've got a 30 page report on your PC. The average printer will take about 17 minutes to print out, during which time your PC is either completely incapacitated or operating far slower than usual.

So,what can you do?

With the MCS X-Buffer, lots.

WHY THE PROBLEM?

A printer operates at a slower rate than a computer, forcing the PC to wait whilst the printing job is completed. And with a plotter producing a complex CAD schematic, that could be several *hours,* not minutes. Fine if you've got time to waste—but most of us haven't.

WHAT IS THE X-BUFFER?

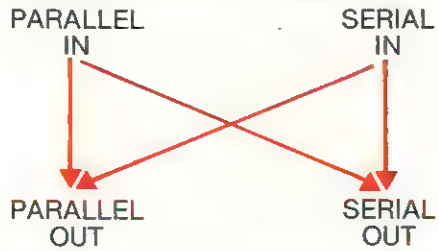
X-Buffer is an intermediate storage device that acts as an extension to the computer's own memory. It allows text or data to be downloaded in seconds and 'held' in the buffer ready to be fed to the printer at the correct rate.

So there's no waiting for printing to finish, the PC is free to be used for other tasks.

In the case of your 30 page report, downloading to the X-Buffer would take around 28 seconds, leaving you 16½ minutes to get on with something far more constructive.

VERSATILITY

Until recently printer buffers had to be specified according to the type of PC and the printer used and the interfaces each had. The X-Buffer is designed for maximum flexibility and incorporates four interfaces: RS232 Serial i/o and Centronics parellel i/o.



The interfaces can be used simultaneously so the buffer can be used in a variety of ways. Two computers sharing one output device (printer, plotter, modem). One computer driving two devices. Two computers using two devices.

EXPANDABILITY

The X-Buffer is also easily expandable by the user, plug-in memory modules offering from 64K to 4 Mbytes of buffer memory.

That's not only good news for users but great news for dealers too—instead of holding large stocks of varying fixed capacity buffer units,the X-Buffer modules allow custom configuration depending on the users' needs.

EXTRAORDINARY VALUE

And the price of the X-Buffer is probably the best feature.

X buffer FROM £159

The 64K basic unit costs a mere £159 RRP including a standard two year parts and labour warranty.

X-Buffer from Micro Control Systems. Don't let your printer (or plotter or modem) keep you waiting.

The X-Buffer is just one of a range of buffer products manufactured and supported in the U.K., available from Micro Control Systems.
Tel: 0602 391 204

micro control systems

BYTECH peripherals
Tel: 0344 427676

AVAILABLE FROM Northamber 01-391 2066
STC ELECTRONIC SERVICES
Tel: 0279 626777

CIRCLE NO. 773

Figure 2 – Listing of SUBS.C

```
/*
Usage: subs inputfile outputfile oldstring newstring [-i]
       e.g. subs old.txt new.txt John Mike
            replaces all occurrences of the word 'John'
            in old.txtwith 'Mike', creating new.txt.
            The -i flag causes case to be ignored.

       Compile and link as follows:
            cl /c subs.c
            link subs+\lib\binmode
            /STACK:48000 /EXEPACK /NOE;
*/

#include <stdio.h>
#include <string.h>
#define TRUE  1
#define FALSE 0

#define MAXLINE 400

char * replaced( char *, char *, char *);

int ignorecase ;

main (argc,argv)
int argc;
char *argv[];
{
   FILE *fopen(), *infile, *outfile;
   char line[MAXLINE];
   char *oldstr, *newstr, *newline;
   ignorecase = FALSE;
   if (argc < 5)
     {
      printf ("\nUsage : subs inputfile outputfile
              oldstring newstring [-i]");
      exit(1);
     }

   if (argc >=6)
      {
       if ( stromp(argv[5] , "-i") )
         {
          printf ("\nInvalid flag %s", argv[5]);
          printf ("\nUsage : subs inputfile outputfile
          oldstring newstring \[-i\]");
          exit(1);
         }
       else
        ignorecase = TRUE;
      }

    if ( (infile = fopen (argv[1], "r")) == NULL ) {
            printf("\nCannot open %s",argv[1]);
                                exit(1);
    }
    if ( (outfile = fopen (argv[2], "w")) == NULL ) {
            printf("\nCannot open %s",argv[2]);
                              exit(1);
    }
       oldstr = argv[3];
       newstr = argv[4];
       while (getline(infile,line,MAXLINE) > 0)
         {
          newline =  replaced (line, oldstr, newstr);
          fputs (newline,outfile);
         }
}

/* Read a line of max length lim, from thisfile */

getline (thisfile, s, lim)
FILE *thisfile;
char s[];
int lim;
{
       int c,i;
       i = 0;
       while (--lim > 0 && (c = fgetc(thisfile)) !=
             EOF && c != '\n')
       s[i++] = c;
       if (c == '\n')
             s[i++] = c;
       s[i] = '\0';
       return(i);
}

/* Return a copy of thisline, with all occurrences of
   oldstring replaced with newstring.  A recursive function,
   this  calls itself until all occurrences of the oldstring
   have been done. */

char *replaced (thisline, oldstring, newstring)
char *thisline, *oldstring, *newstring ;

{
       int i,j,k;
     char leftpart[200], rightpart[200] ,. newline[200];

       i = indx (thisline ,oldstring);

       if (i >= 0)  /* oldstring starts at position i in this line */

              {
                     for (j = 0; j != i; j++)
                           leftpart[j] = thisline[j];
                     leftpart[i] = '\0';
                     j = i + strlen(oldstring);
                     k = 0;
                     while (thisline[j] != '\0')
                           |
                                  rightpart[k] = thisline[j];
                                  j++;
                                  k++;
                           }
                     rightpart[k] = '\0';
                     strcat (leftpart, newstring);

     /* this is the nice bit */
     /* replace oldstring in the rest
        (rightpart) of the string */

     strcat (leftpart, replaced (rightpart, oldstring, newstring));
     return ( leftpart);


              |
       else if ( i <0 )
                     return ( thisline);
/* terminates recursion */
}

/* Check whether t occurs as a substring
   in s and return the position of t in s
   if this is so, else -1. */

indx(s,t)
char *s, *t;
|
       int i,j,k;
       for (i = 0; s[i] != '\0'; i++)
             {
        if (ignorecase)
           for (j=i,k=0; t[k]!='\0' && tolower(s[j])
             == tolower(t[k]); j++,k++) ;
        else
           for (j=i,k=0; t[k]!='\0' && s[j] == t[k]; j++,k++) ;

                     if (t[k] == '\0')
                           return(i);
             }
       return(-1);
}
```

Such a program is shown in Figure 2. It is written in Microsoft C for MS-DOS. The program contains a recursive function (`replaced()`), which does the replacement for one line of the input text. The method used in the function is essentially:

If the text x does not occur return the line unchanged

Otherwise

return the concatenation of the left part of the line (up to the character preceding the first occurrence of x) with y and the result of **replaced()** *called with the remainder of the line as an argument.*

That is to say:

`replaced (leftpart + x + rightpart)`

returns

`leftpart + y + replaced (rightpart)`

Note that rightpart may also contain x as a substring. Figure 1 shows this more clearly.

In `subs.c` I have avoided using any tricky pointer manipulation, so that you don't have to be a C expert to understand it. The program could be improved for efficiency in several ways, but this would also reduce clarity.

It is clear that this method of solving the problem is neat and concise. However, it is also very efficient in execution compared to non-recursive solutions which also operate line by line. The stack overhead is not significant. I have tested it with many occurrences of the string to be replaced in a single line without noticeable degradation.

This is an application from the real world. Perhaps this example, combined with my earlier remarks about embedded semantics, might go some way toward persuading you to consider using recursion more often. EXE

John Healy is a computer scientist at Intepro Systems Ltd, an Irish company which develops automatic test equipment for the electronics industry.

ADVERTISERS INDEX

STOB – I have tested your machine

Tired of reading through those endless, prosy descriptions of the latest PCs?
Ms Stob presents her all-purpose, delete-as-applicable, ozone-hostile PC review.

It's been a long time. The Christmas decorations have gone up *(remember this is to be published in February – Ed)* and the young bud of a bright new decade has flowered since last your jaded scribe took word processor to floppy disk. What transformations will the humble PC undergo during the next ten years? Only time will tell. *(Get on with it – Ed).* This month I will be taking an in-depth look at an important new machine from AFT/Britto-Bulldog Ltd/The Spontaneously Joyful Workers' Collective of the People's Republic of Romania/IDM which has just been rushed to us from Brentford, West London/Catford, South London/Transylvania/the People's Republic of Greenock.

The Super Premium 486/Bulldog Patriot/Sir Nicolae Ceausescu I/IDM PS4 2040 144-23 12A, which we selected as the first review of the year because I sometimes go for a drink with AFT's publicity person/there's a better-than-evens chance that Britto might let us keep the machine/the Publisher hopes to win a lucrative Romanian electrode contract for his Other Business/we always review any old thing IDM may care to toss our way, consists of an ordinary PC/AT sized system unit painted IDM beige/IDM beige/IDM beige/bright red with rainbow spots – ok, IDM beige really, on top of which the monitor, AFT's own colour analogue model/a Natwasukinono Hercules-compatible/a black and white TV set/an optional extra, sits snugly.

I'm notoriously fussy about keyboards, and although I quite liked the feel of this one I was a bit peeved because the space bar was slightly sticky/there was no '£' key/the keys were marked with the Cyrillic alphabet/it's a £200 optional extra.

But what of the machine's performance? I got out our extensive precision-tailored benchmark suite, and after running for several hours I had scored 2,924,700 points with six spare lives/the machine turned out to have the so-called 'Bottom' virus, and began printing VERITY STOB HAS AN ENORMOUS BOTTOM in an endless loop/the machine melted and set fire to my flat/the diagnostic program proclaimed the machine to be completely IDM incompatible. The official technical documentation, which consisted of three loose leaf binders/consisted of a couple of stapled pamphlets printed in Taiwanese/consisted of a crumpled photo of Nicolae Ceausescu awarding himself the Romanian Order of Technological Merit/was not yet available and will be priced from just £45 a manual was no help in this situation, so I phoned up the manufacturers and arranged to meet Sarah down the Dog and Dongle on Friday Evening/got through to someone called Cheryl at the Catford 24 hour Laundrette/got through to a recorded announcement: 'Mr Ceausescu is out, but he will be back soon and it is certainly untrue that 24 years of despotic rule are about to come to a bloody end'/the duty officer refused to speak to me, because he had no authorisation to talk to the press, or, for that matter, any other non-IDM personnel.

In conclusion, I feel it is safe to say that this machine is good/medium/poor/a fitting companion for the IDM PS4 2040 144-23 11A. Can I have my money now please? *No – Ed.*

TAKING THE PLUNGE

Finding a new job ranks alongside Marriage, (or Divorce!) and moving house when it comes to stress. Why not make the process easier by following a few simple guidelines.

- **Motives :** *At the start of the process, it helps to have a clear picture of what is driving your job search. The main forces usually fall into fairly major categories: Money; Opportunity for Advancement; Personality Differences or Travelling Difficulties. Be honest with yourself at the outset and you won't waste time. However, be aware of the changes in your priorities over a period of time.*
- **Markets :** *Have you a clear idea of where you stand in comparison with others of similar skills and responsibilities? The clearer your thoughts on where you want to be, the more positively you can search. Relate your current position to where you want to be. Are your career and personal goals completely realistic? Beware of advertisements touting inflated salaries and try to trade-off factors such as job satisfaction, security and working environment against salary, location and prospects.*
- **Methods :** *Time plays an important role in deciding how best to find a new position. The effort involved in preparing a CV, reading the 'Situations Vacant' columns, mailing, following-up and making interview arrangements, can turn a simple task into a full-time occupation.*

The best vacancies are often unadvertised. As an individual, it would be impossible to effectively cover all the career possibilities available. With professional help, you can widen the scope of your search at no extra cost.

Call agencies by telephone first and gain an impression of their standard of service. At first, they can't tell if you are a client or a candidate. Select a small range of agencies, some specialist to your market and some local to your target area. Some companies use one agency for all their recruitment, so specialists are not always the best.

Above all, don't be railroaded into attending unnecessary interviews. Professional recruitment consultants will only offer advice as to why a particular position is right for you. The decision to attend is yours.

- **Summary :** *Before you begin a job search in earnest, make sure you know why you wish to look. Be honest with yourself and, if you are unsure, seek the opinions of recruitment professionals to use a yardstick. Don't forget that your needs will change over a period of time, so review the situation monthly.*

Know your own worth in the market and whether the move you wish to make is realistic. Choose the method of approach which suits your needs. If out of work, combine your own search with that of specialist and local agencies. If you are busy, remember that agencies can cover a lot of ground for you while working in the background.

Above all, when dealing with agencies and consultancies, remember that it is YOUR needs which come first, NOT their need to fill jobs. So test their level of service before you register.

James Hay, Manager, Recruitment Solutions plc.
Tel: 0753 854256

THE UNIX RECRUITMENT SHOP

X-WINDOWS PROGRAMMER, Cambridge (£Neg)
- Understanding of the X architecture and library
- Systems level experience under Unix with C.
- Experience in interprocess communications.

UPPER LAYER COMMS, Camb, Hemel Hempstead (£13k - £20 + car)
- 2/3 + experience of WAN, OSI, Network protocols.
- Systems design experience under Unix with C
- Session/Transport layer & knowledge of lower layer protocols.

UNIX / C PROGRAMMERS, Basingstoke (£14k - £22k)
- 2/3 + years development experience
- Systems design experience under Unix.
- Programming experience in C or C + +
- Understanding of structured design techniques (object orientated design preferred).

UNIX OFFICE AUTOMATION, Hemel Hempstead (£14k - £22k)
- Broad understanding of Unix applications.
- Programming experience under Unix with C.
- Experience of wordprocessors, databases, device drivers etc.
- Experience in Uniplex, Q-Office, Informix etc

DATA COMMUNICATIONS, Hemel Hempstead (£12k - £18)
- 2 + years development experience
- X25 with broad data communications experience.
- Programming experience in C / Unix.
- Session/Transport layer knowledge would be advantagous.

Contact Tina Brown on 0767 317942 for an informal discussion.

Or write to her at the address shown.

OPEN SYSTEMS PEOPLE

24 Mulberry Close Biggleswade Beds SG18 0HU

WEST YORKSHIRE

Programmers (COBOL, 'C', BASIC and 4GLs), Analyst Programmers, Software Engineers, Project Leaders, Support (pre and post sales), Trainers (MS-DOS, UNIX and Applications). Also Hardware/Software Design Engineers.

FOR YOU NEXT CAREER MOVE AROUND WEST YORKSHIRE:

Telephone Vincent Atherton on Leeds (0532) 504560 or write to:

AIREDALE RECRUITMENT

Realtex House, Micklefield Lane.
Rawdon, Leeds LS19 6AX

AIREDALE RECRUITMENT

WANTED! 'C' MEN & WOMEN REWARDS;

LONDON **£20k-30k**
System Designers required to design top-level specification for state-of-the-art **DATABASE** product including Superior Screen handling facilities. Knowledge of **MS-DOS, 'C'**, with either **OS/2** or **UNIX**. Familiarity with databases such as dBase or Informix useful.

W MIDLANDS **c£18k**
Senior Programmer and Team Leader sought by prestigious publishing house to develop unique Desk Top Publishing system. 'C' expertise essential with any of the following of interest: **OS/2, UNIX, MS-DOS, C++, RDBMS, Presentation Manager, Object Oriented Techniques** or **RISC architecture.**

BERKS **£16k-£25k**
Developer of secure office automation systems seeking exper-ience in one of the following: **'C', SOFTWARE PORTING, LAN/WAN, ISO/OSI, X25, TOKEN RING, UNIX** in a Development, Support, Analysis or Networking role.

HERTS **£18k - £22k**
R&D software engineer required for US software house. 2+ years experience of **'C'** on **PCs. LAN** or **UNIX** desirable, as would be either **Graphics, Financial Applications** or **Assembler**. Possible travel to **USA.**

W. COUNTRY **£Neg.**
Become involved in the development of Process Control Systems across a wide spectrum of hardware, communications and systems. Experience necessary with any three of: **PASCAL, BASIC, MS-DOS, PLC's** or **VAX/VMS.**

SURREY **to £16k**
Expanding software house requires Analyst Programmer to develop user-friendly systems for inexperienced users. **'C'** and **MS-DOS** essential, **80286 Assembler** and **Novell** a bonus.

OXON **to £21k**
Strong believers in the excellence of technology and design for **PC WORKSTATIONS COMMUNICATIONS, VIDEO DISC, CD-ROM, TOUCH SCREEN** and **GRAPHICS**, our client is keen to recruit people with good 'C' experience.

E. ANGLIA **£13k-£18k**
Programming for development and support of **APPLE MAC** and PCs running **PRESENTATION MANAGER** and **WINDOWS**, including PC based **DATABASES** and **INTERNALS**. Experience of **'C', C++** or **PASCAL** on PCs essential.

LONDON **to £25k**
Pioneer in the development of portable computing technology seeking experience of **8086 Assembler, 'C'** and a good degree. Fabulous opportunity to develop skills in object oriented programming and graphical user interfaces.

MANY MORE INTERESTING POSITIONS THROUGHOUT THE U.K!

Contact to find out more about these posts or the many others that we have. As specialists in the industry we can provide you with a professional and confidential service in assisting you with the right career move. Telephone or write to either Antony Bridge M.A., Judy Hortin or Edward Butcher.

01-392 1514

ACUMEN

& SELECTION INTERNATIONAL

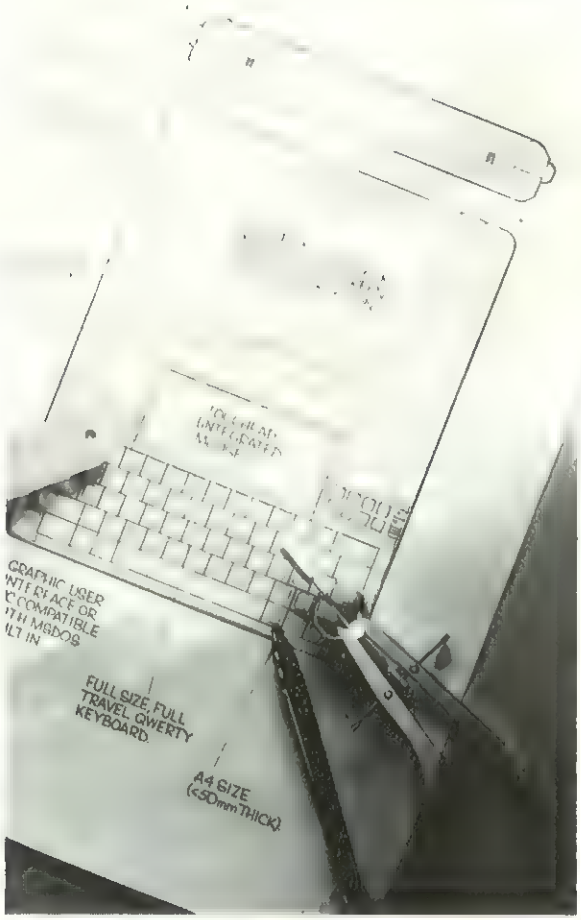
Tideway Yard, Mortlake High
Tel: 44 (0)1 392 1514 Fax: 44 (0)1 392 1500

Evenings till 9pm 01-878 8206

TECHNICALLY SPEAKING

. . . they said it couldn't happen

DEVELOPMENT SPECIALISTS



£15,000 to £25,000

C and Assembler (8086) · CENTRAL LONDON

Well, it has. The Psion MC with its long battery life, light weight and compact A4 size represents the first truly portable computer; its innovative systems software is genuinely multi-tasking and offers a graphical user interface utilising application windows, icons, a pointer device and pull down menus.

Now, in 1990, we are launching a major drive to develop further tools for MC 200 and 400 users and are seeking a number of programming specialists with good C and 8086 Assembler skills to join our applications software development team. This is a unique opportunity to contribute to a software development project being conducted entirely within the UK. Current and future development projects include spreadsheets, word processing, business graphics and voice applications software.

If you have C and 8086 Assembler skills and are a graduate with a good academic record (or can demonstrate exceptional experience), we can provide you with a chance to work within Psion's tradition of successful software development. And in return for your skills and commitment, we will share our mastery of programming for graphical interfaces as well as provide training in Object Oriented Programming methods and techniques.

Opportunities for stimulating projects and career development are excellent because we are continuing to grow rapidly. Talent and effort are rewarded with a competitive salary (inc. free BUPA and a company pension scheme), responsibility and rapid promotion.

To discuss these positions further, telephone our retained consultant, John Carter, on 01-734 4010 (office hours) or 01-741 4087 (evenings/weekends). Alternatively fax your CV to him on 01-734 1297 or write to him c.o. McGregor Boyall, Lyndale House, 49-50 Great Marlborough Street, London W1V 1DB.

PSION

Telephone 01-734 4010

mcgregor boyall

PC Recruitment Specialists

Unique opportunities in

UNIX

Watford To £30k + Car

The Santa Cruz Operation have been leaders in the UNIX software market for almost a decade. Our success is based around one of the world's largest UNIX technical groups whose expertise is renowned throughout the industry. Our continued expansion has created opportunities in several departments.

SOFTWARE DEVELOPMENT To £25k + Car

As a member of the development team you will contribute to providing quality software products for use in all of our markets. The major product development areas include:

✳UNIX ✳NETWORKING ✳GRAPHICS ✳SYSTEM APPLICATIONS

With a minimum of three years' commercial experience, your skills will include in-depth expertise in one of the following:

✳UNIX ✳VMS ✳X-WINDOWS ✳ASSEMBLER ✳'C'

POST SALES SUPPORT To £25k + Car

As a member of the Technical Support Group, you will provide expert advice on all aspects of post sales activity to OEMs, VARs and distributors. With at least two years' experience in development or support of UNIX, your skills will include one or more of the following:

✳'C' ✳NETWORKING ✳DEVICE DRIVERS ✳KERNEL CONFIGURATION ✳X-WINDOWS ✳DATABASES

Whether you see your career developing as a technical specialist or project manager, career opportunities in this expanding team are second to none.

TECHNICAL CONSULTANCY To £30k + Car

Your role will be to provide expert technical advice to major computer manufacturers liaising with product development and support teams to advise on the effective integration of our products. With at least three years' experience in a development or post sales role with a manufacturer, VAR or developer in a UNIX environment, your technical skills will include one or more of the following:

✳UNIX ✳KERNEL CONFIGURATION ✳NETWORKING ✳X-WINDOWS

Finally, the rewards reflect the importance of these positions. Apart from an excellent basic salary you will receive an attractive benefits package, including company car, where appropriate.

For more details, contact our advising consultant,
Mike Williams on 01-495 6686 during office hours, or write to him
at Goodman Graham and Associates, 8 Balfour Place, London W1Y 5RF
enclosing a full CV, quoting reference number 139.

Where YOU can make the difference.

SCO
THE SANTA CRUZ OPERATION

Microsoft

First Class Opportunities in Technical Support

READING, BERKS

Microsoft Corporation is undoubtedly the world's 'Number One' software company marketing a wide range of software products from business applications such as Microsoft Word, Microsoft Excel through languages such as C, to operating systems of the calibre of OS/2.

Since its formation some fourteen years ago the company has played a major role in the development of the microcomputer marketplace showing unparalleled growth worldwide.

Microsoft has in the form of MS-DOS, OS/2, Windows and LAN Manager, set the standards through which the personal computer industry has evolved. At the same time the company has developed mile-stone business software such as Microsoft Excel and Microsoft Word.

In the UK, Microsoft has experienced dramatic growth during the past twelve months and intends to expand still further in 1990. In order to fulfil this aim Microsoft is keen to identify a number of professionals who will complement its existing 'Leading Edge' profile.

APPLICATIONS SUPPORT **£12,000-£15,000 p.a.**

Due to continued success in the field of PC compatible and Macintosh applications, further talent is required to provide Post Sales Support to an ever increasing client base within the Word Processing, Spreadsheet and Graphics User Interface product arena.

In addition to operating at the forefront of Technology, you will promote Microsoft Products through exhibitions and attending Dealer and Distributor training seminars, whilst enjoying a structured, progressive career development path.

SYSTEMS SUPPORT **£13,000-£22,000 p.a.**

You will have a sound technical appreciation of Systems Development principles using OS/2 and/or 'Windows' and 'C'. Primarily supporting development teams within major computer manufacturers, O.E.M's and V.A.R's.

We are also interested in Computer Science graduates, with the drive and potential to succeed in a dynamic environment.

In addition to the salary ranges stated above we offer a comprehensive Benefits Package, including an excellent Company Bonus Scheme.
In the first instance contact our Consultant **Robin Phillips** on **01-388 3818** or **0635 40019** (evenings/weekends), or alternatively send your CV to the London address below.

SCR COMPUTING

Regents Park House, 369 Euston Road, London NW1 3AR

(5466)

BIRMINGHAM 021-631 4030 | LONDON 01-388 3818 | BRISTOL 0272 294195 | BRUSSELS 010 322-640 7151/71

PROGRAMMERS | The Development Company

We are embarking on a number of interesting projects involving real time software on embedded microprocessor systems and microcomputers. We need highly motivated recruits to fill the following positions:

Senior Z80 Programmer ***£14-16k***
2+ years' assembler experience
ICE experience

Z80 Programmer ***£10k-13k***
1 year's Z80 experience
ICE experience.

Senior 68k Programmer ***£14-16k***
2 years' 68k assembler
C experience
PC and MS-DOS experience
8086 experience

If this interests you, call Mike Simpson on 0203 525852, or write to:
TDC, Kings Chambers, Hertford Place, Coventry CV1 3HJ.

C UNIX in the WINDOWS

SEE £20K IN THE BANK!

Yes it is a corny title, but it does sum up our client's main needs and also what your skills could be worth if you can demonstrate expertise in **C Programming** under **UNIX**, ideally including **Windowing** applications. Hardware type is not so important but it would be desirable if you have knowledge of DEC VAX based systems.

In return this leading London-based Systems House can offer you a rewarding workload featuring varied development projects using SSADM and QA techniques, both in-house and at customer sites. Career development opportunities are second to none and typically can lead to Team Leading, Design, Analysis and Project Management, with opportunities to specialise if desired. Benefits are what you would expect from a progressive and professional organisation. (Note, if a London base does not suit you this client would still like to hear from you as a possible for other locations.)

If you feel that your career is going nowhere this opportunity may well be just what you are looking for. Contact JOHN GRAFHAM at Estier Renard Limited for more information, or send (or FAX) your cv to him at the address below. All applications will be treated in complete confidence.

Estier Renard Limited
Human Resource Management
30 Clarendon Road, Watford, Hertfordshire WD1 1JJ
Tel: (0923) 816551 • Fax (0923) 30897

ESTIER RENARD

IOE Lic. No SE 3569

Based in Cirencester, this division of one of the UK's foremost automotive companies is a world leader in electronic diesel engine management systems.

To assist the development teams the company now require

C PROGRAMMERS

This is an opportunity for individuals to use commercial programming knowledge to help in the management of the inhouse UNIX system, whilst gaining experience of more technical applications.

The company offers an excellent salary together with benefits you would expect from a company of this repute which include relocation to the very pleasant area of the country.

In the first instance, please send a full CV to **Mr P M Horton, Impact Technical Recruitment, Calthorpe House, 30 Hagley Road, Edgbaston, Birmingham B16 8QY quoting ref: 241.**

impact
TECHNICAL
RECRUITMENT

A DIVISION OF PACT CONSULTING

'C' PROGRAMMERS

Advanced System Architectures is a company enjoying international renown for its development of technologies and techniques for systems requiring massive computational resources.

ASA is now increasing its system simulation capability to support recent contract awards, and has vacancies for versatile 'C' Programmers to join this team. The successful applicants will be highly conversant with the application of 'C' to time critical applications, and ideally will have some familiarity with simulation systems. The work will be carried out on a distributed Unix environment of Sun workstations backed-up by an Encore 'number cruncher'. Opportunities for working in Ada are also available for programmers who wish to broaden their experience.

In addition to above average salaries all employees of the company enjoy permanent health insurance, private medical cover, a share option scheme, a contracted-out personal pension scheme and life insurance cover. The company is located in very attractive offices adjoining the Alice Holt forest on the Surrey/Hants border.

For further details please contact:

Linda Deakins
Advanced System Architectures Ltd
Tel: 0420 23815

North Block
Bentley Hall
Blacknest, Alton
Hants GU34 4PU

ADVANCED SYSTEM ARCHITECTURES

OPTIONS AND OPPORTUNITIES FOR 'C' PROGRAMMERS

I.T. CONSULTANCY — **SURREY to £25K**

Small but well-established expanding Prestige Co developing high quality custome software, UK & International clients. Tremendous variety, overseas travel, total project involvement, 2 positions calling for experience of C/UNIX on Minis *or* C/MS-DOS on PC's

IMAGING SOFTWARE — **HEATHROW to £18K + CAR**

Development & Support of Sotware Solutions for UK & European Financial Sector customers. American Co, 13 years in UK offering Major Co benefits, International & UK travel & a lot of technology! 2 positions, key skills 'C' & COBOL *or* 'C' & 68K ASSEMBLER.

AIRLINES & TELECOMMS — **SURREY £13K-19K**

Prestige British Co, leading the field in Software Developments for the International Air Travel industry offer range of opportunities (Junior to Senior) in Development, Systems Design & Project Management. Key skills 'C' & PASCAL on VAX (LAN'a/Structured Techniques for senior roles). Occasional travel to customer sites (eg Mexico, Barbados!).

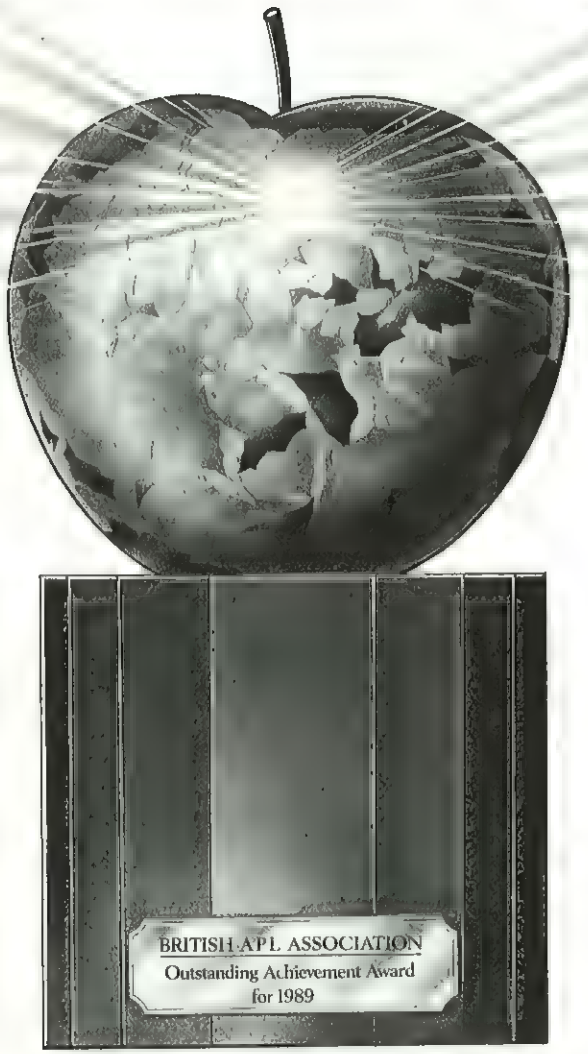
If you're well-qualified and seeking an interesting career move (not just another programming job) - let your 'C' take you places.

Contact George Burrows NOW on 01-549 6441 (24 hrs.)

ab executive

59 Eden St, Kingston-Upon-Thames, Surrey KT1 1BW. Tel: 01-549 6441 (24-hrs)

USE YOUR APL SKILLS WITH AN AWARD WINNING TEAM



British Airways has just won the APL Association Achievement Award for 1989 — a recognition of our contribution to the development of APL for use across worldwide telecommunications networks.

So what better time than now for you to be part of our innovative IT team at Heathrow?

By using advanced technology to collect, process and analyse complex marketing and financial data, you will make a real contribution to the commercial success of British Airways. This involves developing your APL skills on projects ranging from decision support systems for our international sales force to Executive Information Systems for our Company Directors.

ANALYST PROGRAMMER
£14-20K package

With at least 4 years' commercial computing experience, you will ideally have a mathematical or Operational Research background and be familiar with decision support systems. A knowledge of VM/CMS, C, PL/1, SQL or modular programming would be an advantage. Excellent problem-solving skills are essential. Ref: JDW/1906/210.

PROGRAMMER
£10-17K package

With at least 1 year's commercial computing experience, you will ideally have a mathematical or Operational Research background and a knowledge of VM/CMS, C, or PL/1. Ref: JDW/1906/211.

For both posts you will need to be self-motivated and able to work well on your own or as part of a team. There are outstanding opportunities to advance into team leadership, business analysis, or a specialised technical area.

The rewards, like the challenges, are substantial. Apart from a performance related salary, you will receive a profit share, holiday bonus, holiday travel opportunities, outstanding sports and social facilities — and the kind of career development opportunities that only a world leading IT operation can provide.

Interested? Then please write with a full c.v., quoting the appropriate reference number, to: Selection & Assessment, British Airways Plc, "Meadowbank", PO Box 59, Hounslow TW5 9QX.

BRITISH AIRWAYS

TECHNICAL SOFTWARE

Advanced Graphics
M3 Corridor **to £20K**

Head up displays, Image/Signal Processing and Advanced Graphics Engines. Just a sample of the exciting projects available to you within this expanding graphics consultancy and design house. A rare opportunity to expand your graphics experience within the field of commercial and military visual simulation, image processing and graphic control and display systems. If you have real time or high level languages experience and require technical interest and simulation in your next career move then this is the ideal opportunity. **Ref: MJ/64**

C → C ++
Northern Home Counties **to £30K**

Would you like to develop your expertise in Graphics, Database, Data Comms, Object Orientated Design, Expert Systems or Distributed Processing? Would you welcome the opportunity to TRAIN IN C++? If so, and you can offer a good technical degree and a minimum of 1 years experience in C/UNIX you could be working on a variety of technically challenging projects within a small rapidly growing Systems House. Along with this golden training opportunity my client offers Private Medical Care, Company Pension, Bonus Scheme and Employee Share Options. **Ref: MJ/63**

Commercial Communications S/W
Middx. **To £20K**

A prime opportunity to utilise your software experience in the voice or data comms. industries to enter the exciting world of commercial command and control systems. A young British company leading the way in the design and manufacture of flexible communications control systems is looking for young software engineers with 2 years C, YOURDON and ASSEMBLER experience to assist them in developing their products and maintaining their envied market leading position. A rare opportunity to work in a rapidly expanding technically challenging field. **Ref: MJ/62**

Unlimited Career Progression
Bucks **to £27K**

If you are a creative, ingenious software engineer with 3+ years experience in C, UNIX, VAX/VMS and IBM PC our client offers you the chance to work on a variety of challenging and stimulating emulation packages. You will be involved from conception through to completion and work on a variety of platforms including state-of-the-art unreleased workstations. The opportunity and package offered by this young ambitious company is second to none. **Ref: MJ/48**

Software/Hardware Engineers
South Coast **to £25K**

Vacancies exist for experienced Software/Hardware engineers to join a highly innovative high tech. telecomms company. If you can offer PC, ASSEMBLER, 68000 and real time comms experience our client can offer a dynamic environment within prestigious new premises at an idylic location. An excellent salary with first class benefits are available for the right people. **Ref: MJ/61**

Recent Graduates/ MSc Graduates
Various locations **to £15K**

We have a large number of clients ranging from Major International Defence Contractors through to small Independent Software Houses who have an urgent requirement for technically qualified graduates with software experience. If you can offer C/UNIX, ADA, PASCAL or any Real Time experience we can offer you a number of interesting opportunities. **Ref: MJ/57**

For more details on these opportunities call ***Mike Jenkins on 0442 231691*** *office hours or 0582 456417 eves/wkends. Alternatively mail CV to address opposite or fax on 0442 230063*

Executive Recruitment Services

Hempstead House, Selden Hill, Hemel Hempstead, Herts, HP2 4LT

TECHNICAL SYSTEMS DEVELOPMENT

London, Home Counties, South & West

£10,000 to £30,000 per annum

Interactive Resourcing is a specialist recruitment company serving the real time, scientific and technical sectors of the Computer and Electronics industries. Our client companies include large multinationals, systems and software houses, and smaller privately owned organisations. The positions with which we deal provide the opportunity for career development either in terms of technical specialisation, management responsibility or simply in terms of broadening your experience.

OPENINGS CURRENTLY EXIST FOR:

- Development Managers
- Project Managers
- Consultants
- Team Leaders
- Programmers
- Software Engineers

IN A WIDE VARIETY OF AREAS SUCH AS:

- Graphics/Windowing
- Communications/Networks
- SCADA/Industrial Control
- Avionic, Naval, Military C^3I
- CASE/IPSE
- Simulation & Modelling
- Secure Information
- AI/Robotics/IKBS
- Electronic Warfare

YOUR SKILLS SHOULD INCLUDE SOME OF THE FOLLOWING:

VAX	VMS	'C'	ORACLE/INGRES
SUN	UNIX	ADA	OOD/OOP
INTEL	RMX	PASCAL	X.25/X.400/ISDN
MOTOROLA	0S9	FORTRAN	YOURDON/SSADM
PCs	MS-DOS	ASSEMBLERS	FORMAL METHODS

To discuss specific and suitable opportunities please call us for further information on **(0256) 882826,** or forward your CV to us at the address below in the strictest confidence and we will contact you at a time convenient to you.

Interactive Resourcing

Interactive Resourcing Limited
8 Campbell Court Bramley Basingstoke Hampshire RG26 5EG
Telephone: (0256) 882826 Fax: (0256) 882933

'C' PROGRAMMERS

```
# INCLUDE<OPPORTUNITIES.h> */ THE BEST AROUND /*
# INCLUDE<EXPERIENCE.h>*/ 2YRS+ REQUIRED /*
# DEFINE ENVIRONMENT "Development"
# DEFINE PACKAGE "Excellent"
  main
{
printf ("call now or send cv for an early interview/n")
printf ("we meet all relevant candidates/n evenings possible/n")
}
```

Programmers - Any Language — £Negotiable

Are you Graduate or HND/HNC material with formal analysis training (SSADM, CACI, JSM)? Would you like to retrain to AS/400 & RPG? We have excellent opportuniities with unlimited prospects for Analyst/Programmers, so send a Curriculum Vitae or contact Terry for immediate interview.

Technical Support Manager — £Negotiable

If you have a strong UNIX and DOS operating system and applications background, and feel your next move is into management, consider the possibilities with a manufacturer based West London specialising in several vertical markets in which they are distributing their own unique products. The Company offers a generous salary package, company car and substantial benefits. Contact Rob for an immediate interview.

'C' + UNIX/AIX Programmer — to £17k+ bonus

System & Solution Consultancy based West London seek 'C' programmer, preferably with above operating systems. DBase and Accounting module knowledge is preferable, as is an HND or BSc, and 18-24 months' experience. The Client is also seeking to fill a more junior post with the above system, for which training will be given. Contact Terry Nelson.

Programmers & Analyst/Programmers — to £30k

West London Multinational seeks Programmers and Analyst/Programmers for development work. DEC, IBM M/F or S3X experience, minimum 2 years' required. Good Honours Degree Essential.

WE CURRENTLY HAVE SEVERAL WIDE RANGING SUPPORT/DEVELOPMENT POSITIONS,
SO IF YOU ARE SEEKING A NEW APPOINTMENT WITHIN THE INDUSTRY, CONTACT:

Ronicom Recruitment

46-47 Pall Mall, London SW1Y 5JG
Tel No: 01-321 0245 Fax No: 01-839 7629

C SPECIALISTS

'C' Programmers — £12 to 25,000 Central London

This software house are currently seeking programmers with 'C', Windows and mainframe link knowledge to join their team developing Executive Information System (EIS) tools and providing a full consultancy service to their clients. Dealing primarily with corporate clients they seek people with in-depth technical knowledge, excellent communication and presentation skills and a professional and enthusiastic outlook and preferably of degree calibre. *(ref. no. PCEX3/1)*

Software Development Team Leader — £18 to 22,500 City

A market leading software house, developing applications primarily for financial institutions, requires a Team Leader of the highest calibre. Educated to degree level, preferably in a relevant discipline, the successful applicant will possess team leading experience and at least 1-2 years in industry programming in 'C' or a similar structured language. Knowledge of MS DOS, Microsoft C and Networks is highly advantageous. If you are organised, with excellent interpersonal skills, enjoy responsibility, and are keen to work in a company that offers the chance to utilise latest PC technologies, this opportunity could well be the one you are looking for. *(ref. no. PCEX3/2)*

'C' Programmers/Analysts — £15 to 30,000 Nr. Winchester, Hampshire

Our client is a small but rapidly expanding company specialising in Artificial Intelligence. They are looking for bright sparks with at least 2 years solid 'C' experience, ranging from straight coders to senior analysts. Successful applicants will show great potential and be able to work under pressure. Possibly from an engineering background, applicants should be smart and gregarious as visits to client sites will be part of their job. *(ref. no. PCEX3/3)*

C Analyst/Programmers — £14 to 22,000 West London

Moving confidently with the leading edge of software technology, our client is a medium sized software consultancy with an enviable client base. They are now seeking two high quality graduates with good C, Unix and X protocols knowlege. They already have an excellent share of the growing networked Sun workstation market and are determined to enhance their market position. The successful applicant will be determined to keep abreast of rapidly changing workstation trends and willing to contribute to our client's success. *(ref. no. PCEX3/4)*

Telephone 01-734 4010

For further details, please contact Richard Jordan, quoting the appropriate reference number, on 01-734 4010 (office hours) or 01-998 7450 (evenings/weekends). Alternatively, write to him at McGregor Boyall, Lyndale House, 49-50 Great Marlborough Street, London W1V 1DB or fax your C.V. to him on 01-734 1297.

mcgregor boyall

PC Recruitment Specialists

OPEN SYSTEMS.....
.....OPEN DOORS

JOHN BROWN ASSOCIATES is a Computer Recruitment Consultancy. Our specialisation is the recruitment of professionals conversant with the working environments of UNIX, DOS and their related software products, tools and techniques.

Below are a sample of the type of positions we have available. If you would like more information on them or others we currently have available, make a note of the telephone number stated below and give us a call to discuss job opportunities.

C++/UNIX DEVELOPMENT
Hertfordshire - to £20,000

A small but rapidly growing Systems House urgently requires graduate calibre programming professionals who are keen to develop expertise in Object Orientated Programming, Windows, Expert Systems and Graphics.

To qualify you must have at least 12 months programming experience within a 'C'/UNIX environment. (Preference will be given to candidates currently utilising C++, however full training will be provided.)

UNIX SYSTEMS DEVELOPMENT & TECHNICAL CONSULTANCY
London - to £30,000

This market leader in Open Systems Consultancy needs torecruit individuals with commitment to OSI.

The successful candidates will be commercially aware graduates who have significant industry exposure and excellent C programming skills, working knowledge of the UNIX operating system and relational databases. UNIX Kernel level expertise and exposure to Communication Protocols are necessary requirements for the Systems Developers. Applications design and development skills utilising INFORMIX, ORACLE or INGRES within recognised structured methodology environments are necessary requirements for the Technical Consultants.

CAD - Cambridge - to £22,000

Due to the expansion of their software development operation, this manufacturer has a number of positions available for Software Engineers with experience of CAD systems, GIS software and systems tools. To qualify, you need to be a graduate with experience of C, Pascal or Fortran and UNIX or VMS. Extensive involvement with software design, and implementation is a prerequisite and project co-ordination experience will be a distinct advantage.

PRE-POST SALES AND TECHNICAL SUPPORT PROFESSIONALS
£15,000-£25,000 + Car
(Midlands, London, Home Counties)

Are you a Support Professional looking for:

- ▲ **A Change in direction?**
- ▲ **A new challenge?**
- ▲ **A higher salary?**

Do you have experience of:

- ▲ **'C'** ▲ **UNIX?** ▲ **DOS?**
- ▲ **4GLs and Relational Databases?**

If you have answered yes to any of the questions posed or are unsure of your correct answer, why not give us a call to make an initial enquiry about positions currently available.

John Brown
associates

S O F T W A R E S E C U R I T Y

With £1 Billion Worth Of Protected Software...

Rainbow Is The Safe Bet.

SentinelPro™

• Runs under DOS, OS/2 and Xenix • Algorithm technique (Never a fixed response) • External parallel port installation • Minimal implementation effort • Higher level language interfaces included • 100 times faster than fixed-response devices (1 ms) • ASIC design for reliability

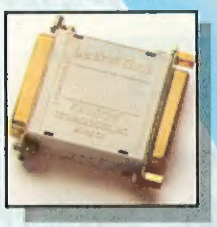
Sentinel-C™

• Protects multiple packages with one device • 126 bytes of non-volatile memory programmed before shipment of the software • Rainbow supplies a unique adapter for programming the unit • Higher level language interfaces included • Runs under DOS, OS/2 and Xenix • External parallel port installation

Eve™

• For the Macintosh SE and II • Complies with Apple Desktop Bus Interface requirements • Rainbow-assigned developer passwords to prevent tampering by other developers or sophisticated "hackers" • 7 locks per key, usable individually or in combination, on one or up to seven applications

SentinelShell™



• Runs under DOS on IBM PCs and compatibles • Protects without requiring access to the source code • Completely transparent to the end user • User-friendly software • Pocket-size key attaches quickly to any standard PC parallel port • ASIC design for reliability

DataSentry™

• Completely user-installable • Pocket-sized external device • Menu-driven, user-friendly interface • Single- or multi-user security system • Audit trail, log-on identifiers and automatic encryption/decryption of entire directories • Secures data transmitted by modems • Prevents recovery of data by utility programs

Software developers creating the latest applications for the IBM PC/XT/AT, PS/2 and compatible systems can now turn to the Software Sentinel range of hardware keys for the *first-class* in *world-class* software protection.

There's the best-selling SentinelPro, known worldwide for its virtually unbreakable security, its ASIC technology and its invisible operation.

A close relation, the Sentinel-C for custom configurations, enables multiple package protection with a single device.

For the Apple market, security-minded Mac software developers can now secure their return-on-investment, too. Eve plugs into the Mac ADB connector and is completely transparent to the user–providing up to seven programmable security locks per key.

Rainbow's latest protection strategy is the SentinelShell–that lets users place a "shell" around existing, off-the-shelf programs. Access can be limited to those issued a key, so that libraries, educational establishments and corporations can very simply guard their investments.

Available soon from Rainbow is the DataSentry, a user-installed key that provides low cost security for sensitive data in both database applications and corporate/banking environments.

No matter where you sell your software worldwide, stay in control of your distribution and revenue by choosing the internationally accepted standard in protection... Rainbow Technologies. Be sure. Protect your pot of gold at the end of the rainbow.

CALL TODAY OR ORDER NOW.

☐ Please send me a SentinelPro Evaluation Kit.
I enclose £50 + VAT Payable to Rainbow Technologies Ltd.

☐ Please debit my credit card. Access ___ Visa ___ Amex ___
Expiry Date _____ (PLEASE TICK)

☐ Please send me more information.

Name _____

Position _____

Address _____

Telephone _____

Signature _____

RAINBOW TECHNOLOGIES LTD., Shirley Lodge, 470 London Road, Slough, Berkshire SL3 8QY.

RAINBOW TECHNOLOGIES

Shirley Lodge, 470 London Road
Slough, Berkshire SL3 8QY
Tel: 0753-41512 Fax: 0753-43610

CIRCLE NO. 775